コンピュータ・グラフィックスとCAD/CAM/CAE雑誌ピクセル　1989年2月1日発行/毎月1日発行/通巻77号/1984年5月2日第3種郵便物認可　ISSN 0287-511X

PIXEL

'89 2

No.77

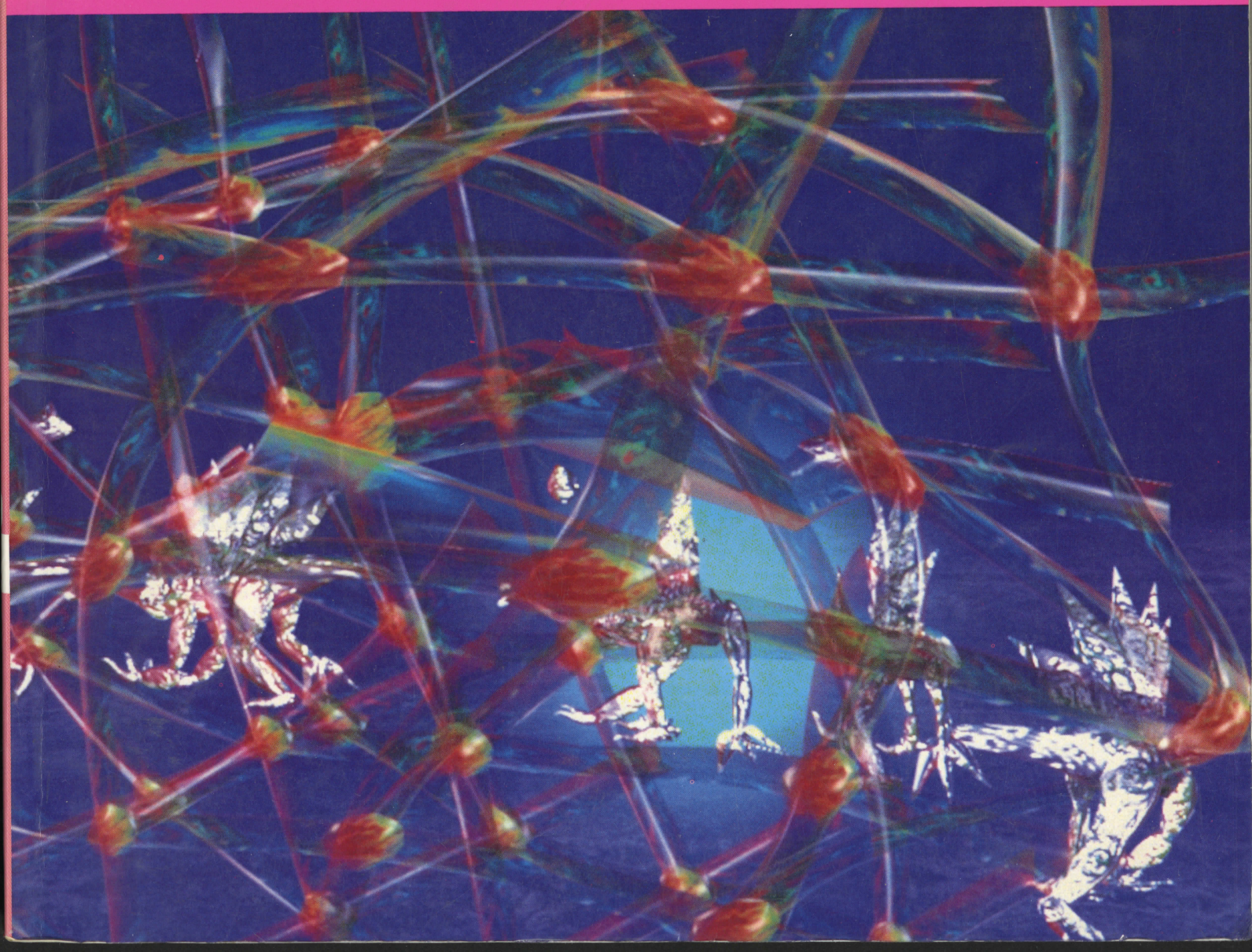
特集 新しいCG表現を可能にするメタボール

特集 3次元機械設計用CAD/CAM技術と使い方

特集 CADシステムのカスタム化

★テクニカル・イラストの変遷とCAD化への可能性

★ここまできたリアリズム——四半世紀の進展



Good News!
3Dグラフィック
デザイン・クリエータ

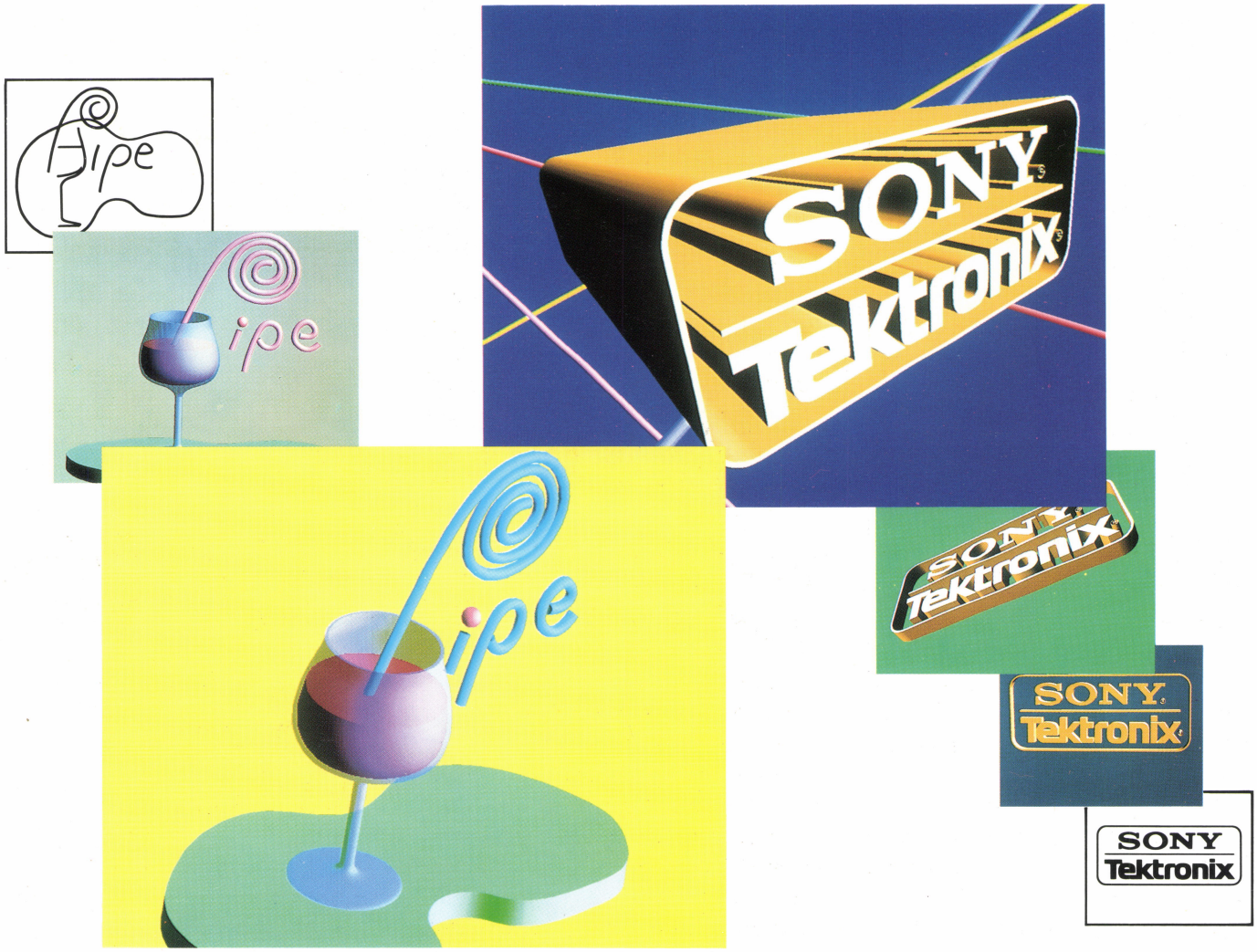
SONY Tektronix ●グラフィック世界のリーダー

ソニー・テクトロニクス

ソニー・テクトロニクス株式会社
情報機器部／東京都品川区大崎1-6-4 新大崎勧業ビル 〒141
TEL03・779-7611
仙台営業所／TEL022・267-2181　土浦営業所／TEL0298・24-2602
名古屋営業所／TEL052・581-3547　大阪営業所／TEL06・947-0321
広島地区／TEL082・247-0661　福岡営業所／TEL092・472-2626

創造性躍動
2次元グラフィック・デザインを簡単に3次元表示に

●ロゴタイプ、シンボルマーク、ピクトグラフ、パターン ●アニメーション



線画やマークを立体表示/3Dグラフィック・デザイン・クリエータは手描き線画、ロゴタイプ、シンボルマーク、ピクトグラフ、パターン等の2次元グラフィックを元にして任意の奥行きを与えるだけで瞬時に3次元表現化を行えるシステムです。

立体モデル、ろくろ機能、パイプ機能でリアルな表現に/基本的なモデリング機能で2次元グラフィックから球、円錐、円柱、ドーナツ型、直方体等のきれいな3次元形状化がリアルタイムで行えます。さらにろくろ機能によりなめらかな回転体を作成したり、曲線を自由にたどれるパイプ機能等があり、リアルな表現が得られます。

クリエイティブ表現/出来あがった3次元表示は拡大、縮小、回転、移動、コピー等自由自在。組合せも自由ですのでクリエイティブな3次元グラフィック・デザインを作成できます。

半透明表示も行える豊富なカラー/カラーは管面のメニューを見ながらインタラクティブに色相、明度、彩度を選びます。豊富なカラーに加えて半透明表示も行えます。

リアルなライティング/ライティングは最大5光源を設定できます。周囲光、拡散光の強弱や方向も思いのままに設定できます。

アニメーション作成/簡単な操作で拡大/縮小、回転、移動を利用して自動補間によるなめらかなアニメーションを作成できます。当社の3Dアニメ・クリエータとの組合せでビデオ・アニメーションも短時間でできます。

新　製　品

3Dグラフィック デザイン・クリエータ

（資料請求番号　表2）

D-SCAN

テクノロジーを彩るフルサイズ、フルカラー。

広がり続

A0~A4・Auto-Cut　4096-Color

16-dot／㎜の高解像度で4096色のフルカラー出力。世界初のA0～A4サイズ・オートカット連続出力。タイミングマーカの検出による用紙走行、巻戻しの高精度化による色ずれのない鮮明な出力。クラス最高38㎜／秒の実速を誇る用紙送り速度。データ処理速度5000ベクタ／秒以上、他に類のない64Mdot／秒以上のラスタ化速度。イーサネット、チャネルなどの豊富なインタフェースなど、エンジニアリングニーズを満たす高機能を標準で装備した国産初のカラー静電プロッタEP-4010が、待ち望まれたフルサイズ、フルカラー、フルパフォーマンスに対するD-SCANからの解答です。

けるエンジニアリングに、創造力と表現力の大幅なパワーアップを実現し、膨大な情報を含む長尺図面からA4サイズまでの色ズレのない高画質カラー出力を自動レイアウト・オートカットで今までにない前面出力するEP-4010。さらに、ラインはもとより、塗りつぶし時の重なり部分を明確に再現する後書き優先機能、日本で欠かせないJIS第1・第2水準漢字のサポートなど、日本のエンジニアとエンジニアリングの隅ずみまでを熟知した、CGの総合ブランドD-SCANならではの実績とテクノロジーが、カラー静電プロッタを開発・設計の戦力として具体化しました。

■EP-4010の主なパフォーマンス■自動レイアウト機能、オートカッタを標準装備して、1ロール長(150m)の連続出力を実現■オペレータの操作性を重視した用紙の前面セット■イーサネット、チャネルなど豊富なインタフェース■コントローラを内蔵したコンパクト・軽量設計■ラスタメモリユニット(オプション／最大140MB)による大量ベクタデータへの対応■LCDパネルを採用した優れたオペレータインタフェース機能■普通紙、トレーシングペーパーなど豊富な用紙種類に対応■ホスト、ローカルで可能なカラー／モノカラー出力の切り替え■EP-2000シリーズで培ったノウハウによるハイコストパフォーマンスの達成など、カラー出力を使いこなせる身近なテクノロジーとして提供しています。※イーサネットはXEROX社の商標です。

OEM・販売代理店　募集中

カラー静電プロッタ NEW EP-4010

SEIKO I

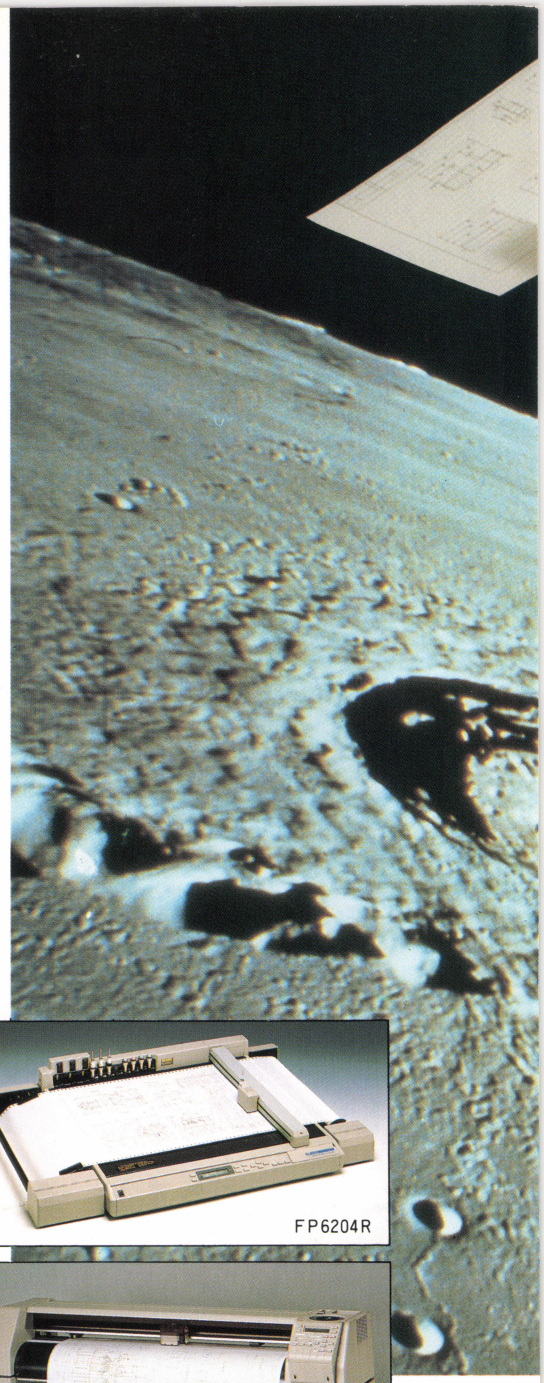
セイコー電子工業株式会社
情報関連事業部

本社:東京都江東区亀戸6-31-1 〒136 TEL:03-684-5711
水戸:0292-26-5701／名古屋:052-731-3181／大阪:06-305-6002／広島:082-263-6260

（資料請求番号　1）

無人連続作図

ARFシリーズ登場



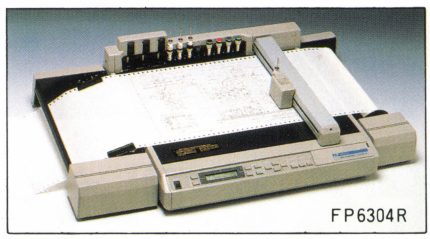
プロッタの新しい可能性が、いま 拓かれました。

40mの記録紙を自動給紙するオートロールフィーダを装備した省力化プロッタ「ARFシリーズ」。最上位モデルには鉛筆芯80本を自動給排する〈CLEARシステム〉を搭載し、初めて鉛筆での大量無人連続作図も可能となりました。作図効率の向上、夜間自動作図での省力化などCADの本格稼動に向けて、限りなく可能性が拓かれてきました。

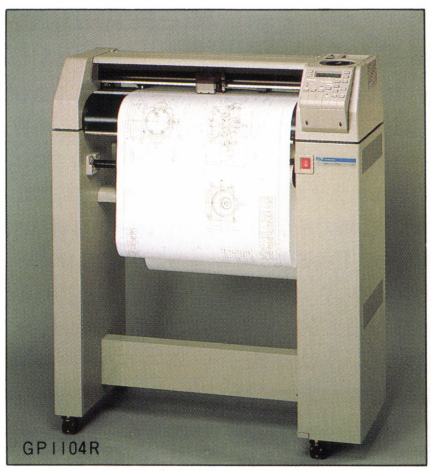
(40mロール紙を使用時、A0で33枚、A1で44枚、A2で57枚、A3で88枚の連続作図が可能です。)

さらに高速、さらに高描画。

無人連続作図プロッタ「ARFシリーズ」は、A0タイプからA3タイプまで11機種。用途に合わせて多彩にお応えいたします。いずれも16ビットCPU搭載、デジタルサーボ駆動方式の採用により処理速度も一段と向上。また作図順序を効率的に並べかえたり(ベクトルソーティング機能)、ペン交換時間を削減する(ペンソーティング機能)最新のインテリジェント機能の搭載によって当社従来機種と比べて作図時間が半減いたしました。


FP6304R

FP6204R


GP1104R

GP1004R

グラフテック株式会社

本社/〒108 東京都港区三田3-13-16 三田43森ビル16階
●カタログのご請求は、本社広報宣伝部まで。
お問い合せは●プロッタ事業部 TEL.0466-87-4111(代)

●北海道営業所 011-232-1332 ●東北営業所 022-236-3405 ●関東営業所 0485-25-2381 ●茨城営業所 0298-58-1211
●東京営業所 03-444-7011 ●公需営業所 03-444-8281 ●秋葉原営業所 03-834-5688 ●多摩営業所 0423-25-5050
●神奈川営業所 045-541-6811 ●厚木営業所 0462-22-4971 ●中部営業所 052-776-0821 ●関西営業所 06-305-1381
●西部営業所 082-261-2931 ●九州営業所 092-474-2441

(資料請求番号 2)

A0タイプ	GP1004R(CLEARシステム搭載)	¥2,450,000
A1タイプ	GP1104R(CLEARシステム搭載)	¥2,150,000
A2タイプ	FP6201R	¥1,460,000
	FP6203R	¥1,510,000
	FP6204R(CLEARシステム搭載)	¥1,660,000
A3タイプ	FP6301R	¥ 660,000
	FP6303R	¥ 710,000
	FP6301T(オートカッタ付)	¥ 750,000
	FP6303T(オートカッタ付)	¥ 800,000
	FP6304R(CLEARシステム搭載)	¥ 860,000
	FP6304T(CLEARシステム・オートカッタ付)	¥ 950,000

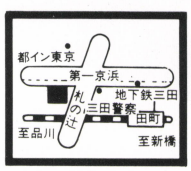
※価格はいづれもRS-232-C・セントロニクス両用機。

鉛筆芯80本分の連続作図を実現した CLEARシステム

鉛筆作図のオートメーション化を推進するNEWテクノロジーです。8本の多芯鉛筆プランジャに、それぞれ10本の芯を挿入。それぞれが芯の消耗を検知し、自動的に交換していきます。鉛筆プロッタの理想を求めたメカニズムCLEARシステムです。

高速性・高描画品質に差がでる 駆動方式

デジタルサーボ駆動方式はDCモータ制御により、〝高速〟〝高描画品質〟〝低騒音〟が可能となりました。またフィードバック機能により、〝より正確〟な駆動が得られるこの方式は、グラフテックが他社に先がけて標準化したひとクラス上を行く高性能駆動方式です。



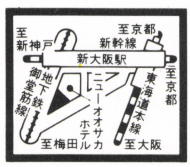
見て、触れて、体験できるグラフテック総合ショウルーム

PLAZA TOKYO/OSAKA

プラザ東京 月～金 AM9:30～PM5:30 TEL.03-453-0512
東京都港区三田3-13-16三田43森ビル16F(本社内)
JR田町駅、都営地下鉄三田駅下車 徒歩5分

プラザ大阪 月～金 AM9:30～PM5:30 TEL.06-305-1381
大阪市淀川区西中島5-14-5新大阪INビル8F(関西営業所内)
JR新大阪駅、地下鉄御堂筋線新大阪駅下車 徒歩2分

電話によるご相談は 03-453-0512 インフォメーションセンター



ペンレスプロッタ テクニカルセンター
横浜市港北区新横浜3-18-14光正第1ビル6階
TEL 045(474)2828

CADテクニカルセンター
東京都港区港南2-13-31品川NSSビル6階
グラフテックエンジニアリング内 TEL 03(450)0511

資料請求券 ピクセル 2月号

（資料請求番号 3）

MUTOH

みえてくる、明日のかたち

速さを追求すればプロッタは線から面へ。
MUTOHのラスター・プロッタ――――今、時代の一線へ。

ひとつのイメージを表現するのにも、表現手段はひとつだけではありません。その場の状況に応じて、最適な手段を選択できるのが理想です。そのためには、ひとつの技術にこだわるだけでなく、技術の幅を広げることも大切。そこで、ラスター。ラスターの技術がプロッタに未知の〝速さ〟をもたらしました。いってみれば、線ではなく面で描くプロッタ。図面情報の量にかかわらず、どんな図面も驚くほどすばやく出図します。MUTOHのプロッタはいま、超・線・力。線を超える力をめざします。速く、もっと速く。プロッタはラスターで加速を始めました。

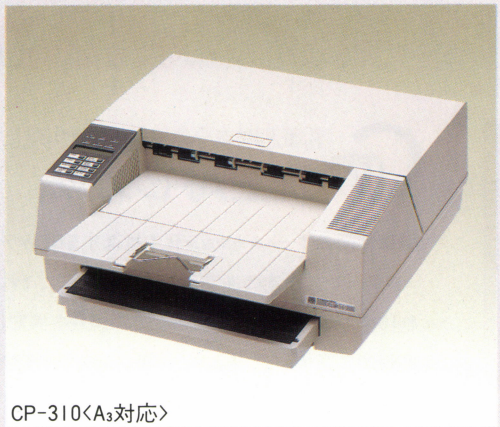
ラスターの性能を身近にした、VERSATEC8500そしてCP-310。VERSATEC8500シリーズは従来コスト1/2のローコストを実現した本格的汎用静電プロッタ。その出図速度はA1サイズで1秒間に最大2.54cm。また記録紙にはロール紙を対応し、長時間連続自動作図を実現。無人化運転も可能にします。さらに1インチ200ドットの解像力にくわえ、自由にセレクトできる線幅、ハーフトーン、塗りつぶしなど豊かな表現力を身につけました。CP-310ではA3で15秒の高速出図を可能にし、しかも従来機種の1/2と大幅にプライスダウン。そのうえPPC用紙がそのまま使用でき、ランニングコストの削減も実現しました。また解像力1インチ400ドットの美しさにくわえ超軽量、さらにペンプロッタとのコンパチブルを可能にするなど、これがまさに国内初のA3スタンダード・レーザービームプロッタの実力です。

超・線・力

VERSATEC High-Average Electrostatic Plotter 8500SERIES
高速・高機能静電プロッタ

VERSATEC
8536〈A_0対応〉

High-Resolution Laserbeam Plotter *Professional Plotter* CP-310
高解像レーザービームプロッタ・CPシリーズ



CP-310〈A_3対応〉

あなたにすばやく応える技術。
Expert-ner
MUTOH PLOTTER

●カタログを差し上げます。ハガキに希望製品の資料請求券をお貼りのうえ、下記住所総合企画室までご郵送ください。

資料請求券
PX-I
VERSA・CP

武藤工業株式会社　東京都目黒区中目黒4-6-1 〒153 TEL.(03)760-6111(代)

技術者募集中 お問い合わせ 人事部 TEL.(03)795-0111

●小山久美

スターダンサーズ・バレエ団在団。

慶大卒業後、文化庁在外研修員として2年間米国留学。

メリッサ・ヘイドン女史に師事。

1年間在団したタンパ・バレエでは、〝パキータ〟のソリスト等を踊る。

●

「幅のある表現力を身につけたいから、

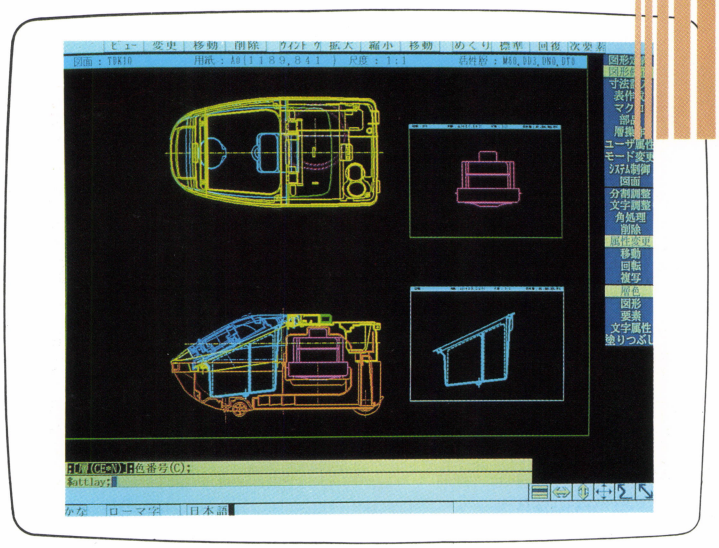
ひとつの技術をマスターしたら、

また新しい技術に挑戦するんです」

（資料請求番号　5）

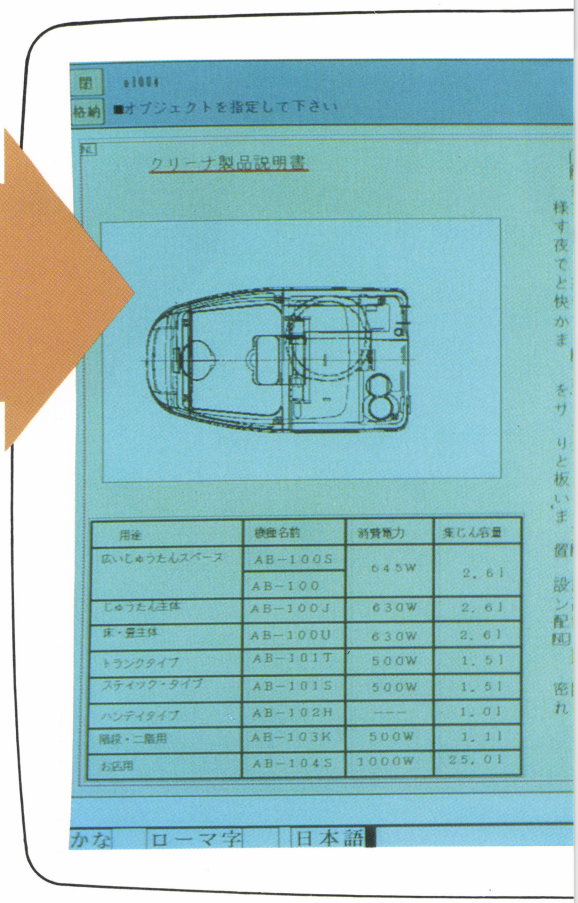
技術の日立

これが、2050Gの



HICAD/W

●マルチビューポート機能を利用して、複数図面の同時操作が行えます。
●業務の効率を向上させるユーザインタフェースを用意しています。
●定形的な一連のコマンド列を登録できるマクロコマンド機能を備えています。



CAD図面と文書データのドッキング

エンジニアリング業務OA化へ、最短ルートで到達できます。

ホストコンピュータに収められたデータを分析して、市場ニーズに合った製品をいちはやく企画する。実際に設計し、解析・シミュレーションを行う。それを製品企画書・説明書としてまとめあげる…。2050Gでは、このような一連の作業をすべて画面上で行うことができます。本格的な2次元設計・製図システム〈HICAD/W*〉や、表やグラフ、文章と〈HICAD/W〉で作成した図形データを組み合わせた高品質な技術文書が完成できる〈エンジニアリング支援ライブラリELシリーズ**〉を用意しました。また、さまざまな解析ソフトウェアやAIツールなどにより、エンジニアリング業務のOA化を支援しています。

*Hitachi Computer Aided Design System/Workstation
**Engineering Support Library

●日立トークンリングネットワークTR4やWS-NETにより、ホストコンピュータとの強力な連携を実現する垂直・水平分散ネットワークを構築できます。日立クリエイティブワークステーション2050/32、2050との混在も可能です。●ソフトウェア開発支援ツール、AIツール、OAソフトウェアなど多彩なソフトウェアを利用することができます。また、解析ソフトウェアも豊富に利用できます。

株式會社 日立製作所 ●お問い合わせは=情報システム営業本部 電話(03)763-2411 または最寄りの支店へ…北海道(011)261-3131・東北(022)223-0121・横浜(045)664-1521・北陸(0764)33-8511・中部(052)562-1111・関西(06)261-1111・中国(082)223-4111・四国(0878)31-2111

HITACHI

簡単ドキュメンテーション。

表やグラフ、CAD図面、文章などを、用紙上に自由にレイアウトできます。完成した文書をこの状態で修正することも可能です。

ELシリーズ

- 文章・図形・グラフなどの混在する文書を、画面を見ながら容易に作成・編集できます。
- 高度な文書管理機能を提供します(章・節管理など)。
- ユーザアプリケーションソフトウェアと連携させることができます。

日立エンジニアリングワークステーション 2050 G

HUMANICATION

・九州(092)741-1111 ☆詳しい資料をご希望の方は、お手数ですが資料請求券をはがきに貼り、貴社名、所在地、電話番号、ご氏名を明記して右記へお送りください。〒140 東京都品川区南大井6-27-18 ㈱日立製作所 コンピュータ事業部 宣伝グループ

資料請求券
PIXEL
1/15
2050G

見えないものが、見えてくる。

ビジュアリゼーションの最先端

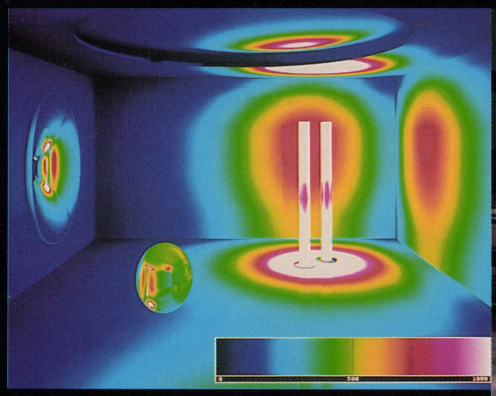
双方向高速光線追跡法
TURBO BEAM TRACING

ターボビームトレーシング(TBT)は、インテグラ独自の双方向光線追跡法により完全大域照明モデルを実現。照明効果を完全に考慮した迫真の画像を生成する、高度CADシステムのための画像生成ソフトウェアです。

従来のレイトレーシング：直射照度のみ

TBTによる拡散光を考慮した相互反射：間接照明が可能

資料提供：
- 建築モデル
  ㈱大成建設
- 照明器具配光データ
  東芝電材㈱
- インテリア
  ㈱フロス



照度分布の色出力

TBTは、特定のCADシステムとのインテグレーションだけでなく、お客様が現在御利用中のCADシステムを機能補間する形でも利用いただけます。

〔業務内容〕

＊照明シミュレーション
＊測定・解析結果の視覚化(照度、温度、電磁波等)
＊CAD/CAMシステムのインテグレーション

その他、科学、技術分野における高度に専門化されたCGシステムの開発、インテグレーション、カスタマイゼーション等。

株式会社インテグラ
〒170 東京都豊島区駒込2－7－11 比米ビル2F
PHONE:03(949)6620㈹6640 FAX:03(949)6640

(資料請求番号 8)

TOYO

またひとつ、理想へ。——TOYOの熱転写方式フルカラービデオプリンターTPG-4300、先進の機能とゆたかな表現力をフル装備して、待望の新登場！ コンピューターによる画像処理、3次元 CAD / CAE システムなどを中心に、研究開発やビジネスのさまざまなシーンで活躍します。

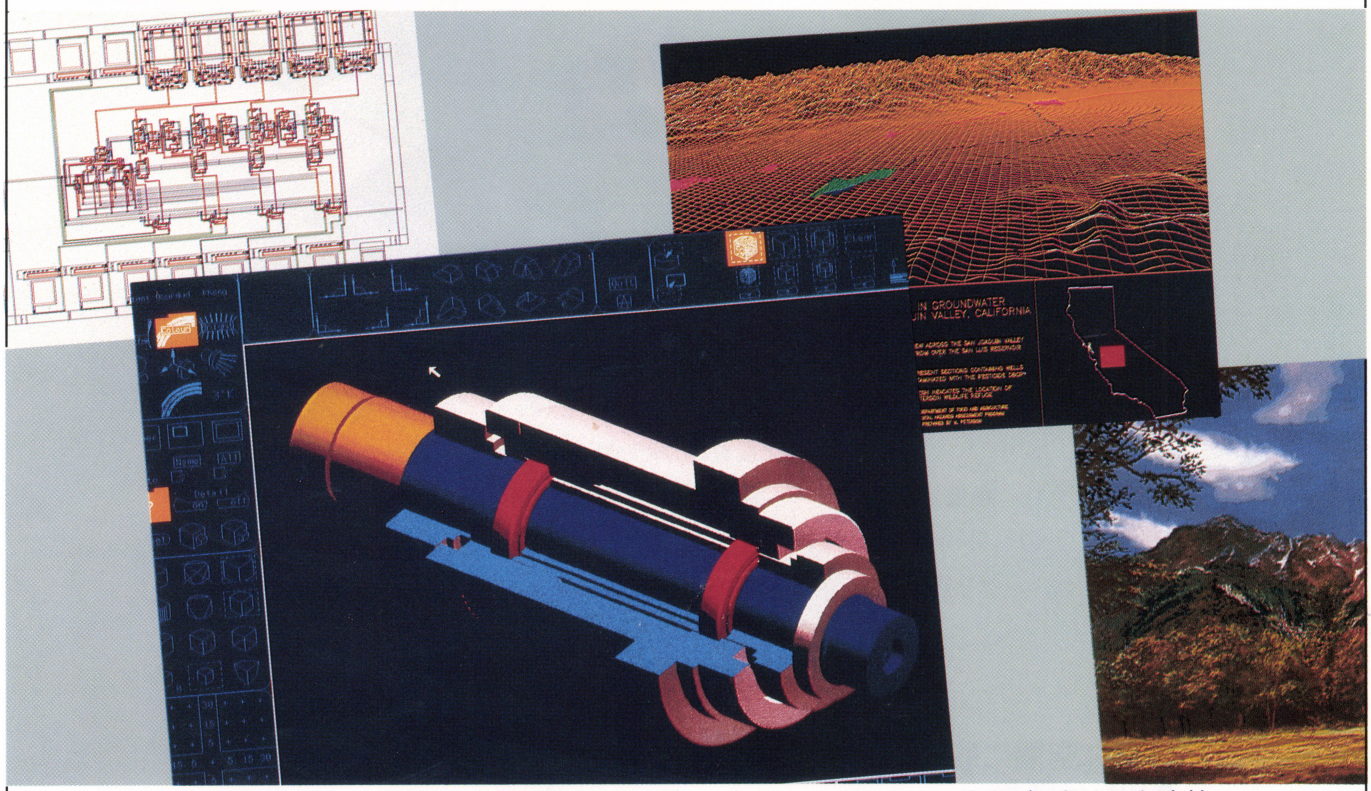
300dpi・A3サイズ 低価格（198万円）

高機能ビデオプロセッサー搭載

3.75MHzから125MHzの高周波RGBビデオ信号まで、短時間で容易に接続できる新開発の広帯域ビデオプロセッサーを内蔵！その安定した性能は、1280X1024の高解像度グラフィックターミナルとも信頼度の高い接続を可能とし、ホストコンピューターをソフト開発負担から一挙に解放しました。

3次元の世界をゆたかに表現

世界に誇るビデオ接続機能と、洗練されたインク特性および高度な印字ヘッド制御技術とのマッチングにより、他には類を見ない光沢と深みのある高品質3次元ハードコピーを供給します。



TOYO TPG-4300

熱転写方式フルカラー

ビデオプリンター

300dpiで印字ズレ色ムラを追放

サーマルプリンターとしては最高の300dpi（11.8本/mm）高密度印字ヘッドと精密なプリンター機構が印字ズレ色ムラを一掃、緻密なワイヤーフレーム画像や高密度電子回路映像をも鮮明に再現します。

豊富な特殊プリント機能

世界初の0.1倍ステップ拡大機能はプリント画面の大きさをXY方向それぞれに1.0倍から16.0倍まで任意の大きさに拡大でき、どんなサイズのディスプレー画面でも常に記録紙面いっぱいにハードコピーします。

その他白黒反転、トリミング、自動センタリング、プリント位置任意指定、カラーアサイン、カラーインテンシティー調整、YMC個別プリント、左右反転、コピーモード等各ユーザーの用途に合わせた使い方が自由自在。1台で8台の種類の異なるEWSを接続できるリモート機能付きマルチプレクサーほかシステム化対応付属装置も豊富です。

株式会社 東陽テクニカ CVP販売担当
〒103 東京都中央区日本橋本石町1-1-2 TEL 245-1351 FAX 271-4757

資料請求券
TPG-4300
PIXEL

OMRON

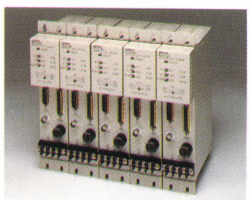
あらゆるアングルで、小さな被写体をとらえる。

ひとつの物体を複数のカメラで同時に撮影し、
画像処理したい——そんなご要望にお応えするのが、オムロン
のヘッド分離型シャッタービデオカメラです。監視や検査工程において、
取りつけスペースを気にする必要がなく、小さな空間で小さな被写体を撮影することができます。

小さな空間に取りつけできる ヘッド分離型

ヘッド部だけを必要なアングルに取りつけることができますので、小さな空間でも複数のカメラで撮影することが可能です。

スリムなカメラコントロールユニット



スリムなブックタイプですので、小さなスペースに並べて設置できます。また、操作部・結詮部を全面に集中させていますので、操作が容易です。

外部シャッタートリガー機能

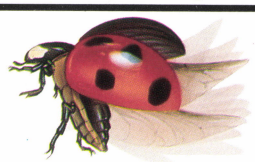
ウィンドゥ設定において、被写体の識別部位を画面上の定位置に確実に収めることができます。

ACアダプタ内蔵

ACアダプタ内蔵により、AC100V電源がそのまま使用できます。

●1/700～1/5600secの7段階の速度設定が可能 ●外部同期モードも可能 ●フレームトランスファ方式採用 ●交換レンズの豊富なCマウント方式採用 ●ストロボなどの光源は不要

CCDシャッタービデオカメラ 形PJLG-60



立石電機株式会社
情報システム統轄事業部／画像システムプロジェクト
〒541 大阪市東区南久宝寺町4-39 御堂筋センタービル
TEL 06(282)2643

技術面でのご質問は、上記へお問い合わせください。

（資料請求番号 10）

Nippon DataGeneral

インテグレーション■統合化への確かな解答

誕生、設計システム構築の新技法。

3次元図形と属性データを一元管理する設計データベース・システムを、日本・データゼネラルは提供します。

INFLAME THE IMAGINATION

統合型設計システム構築ツール

TEO/3D

DESIGN
DATA MANAGEMENT
DISTRIBUTED ARCHITECTURE

TEO/3D™は、これまで個別に管理されていた3次元図形と属性データを同一のデータベースで一元管理。しかも高速応答を可能にした先進の設計データベース・システムです。チーム開発に必要な設計データベースを容易かつローコストで構築し、次世代CIE/CIMシステムのための統合環境を実現。また日本・データゼネラルのAI技術との融合により、知的CADシステムを構築することもできます。

TEO/3Dの主な特長

- 最新のリレーショナル・データベース技術と3次元設計技術を統合。図形/文字データを一元管理するエンジニアリング・データベースを実現。
- 3次元グラフィックス/図形作成ツール、イメージ表示ツールなどをサポート。
- 業界標準言語で記述されたユーザ・プログラムを統合するアプリケーション・インタフェースをサポート。
- 分散型開発を実現するネットワーク機能。

TEO/3Dの主な適用分野

自動車、造船、プラント、建築/土木、環境開発および流通業における情報サービス等。

写真：DSファミリーにて稼動するTEO/3D

TEO/3Dは、データゼネラル社の商標です。

日本・データゼネラル株式会社

本　　社：〒105 東京都港区虎ノ門4-3-13 ☎ 03(438)3611(代)
工　　場：〒361 埼玉県行田市持田2165 ☎0485(54)7161(代)
国内拠点：札幌/仙台/筑波/名古屋/京都/大阪/広島/福岡
海外拠点：アメリカ/カナダ/イギリス/フランス/イタリア/オーストラリア/ホンコン他53ヵ国

（資料請求番号　11）

描いて、切る。一台2役。
高速描画・高精度カッティングのハイコストパフォーマンスを追求したフラットベッドプロッターdrastem 2000シリーズはAO判及びA1判の2タイプ。
●抜群な処理スピード――16ビットデュアルCPUの採用により高速製図を実現。最高速度24m/min・最大加速度0.4G。
●自由曲線が高精度高速カッティング――同時3軸制御方式採用で、自由曲線においても最高レベルの高速カッティングが可能。
●ペンだまりの大幅な改善――ペンヘッドのソフトランディングを採用。始点から終点までの均一な描画によりペンだまりの大幅改善を実現。
その他豊富なインテリジェント機能で対応します。
drastem 2000シリーズ
東洋電機
drastem DIVISION
〒108 東京都港区三田3－13－16 三田43森ビル7F
PHONE 03(457)0741代 FACSIMILE 03(457)0230
曲線をスムーズにリアルカッティング。

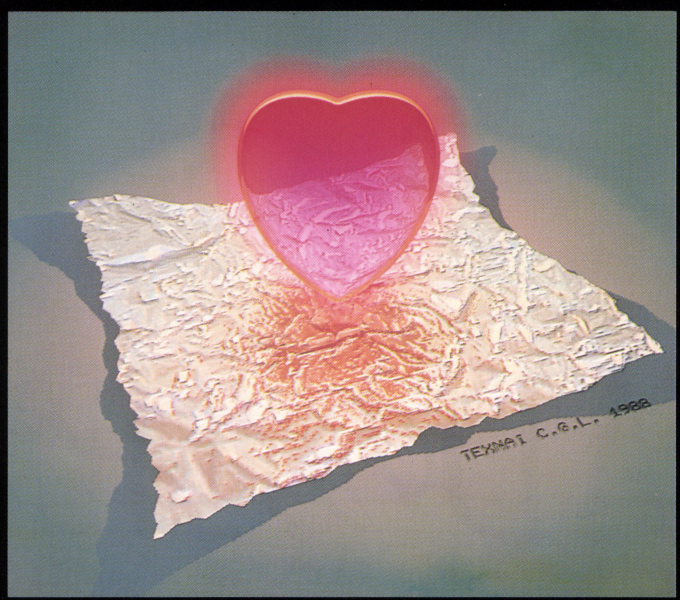
（資料請求番号　12）

TEXNAI
The Quest ... Art & Technology

Τεχηε

［テクネ］

〝Technique〟の語源をなす古典ギリシャ語。
模倣技術としての芸術やその他の諸技術を統一的に表現した言葉で、
建築、絵画をはじめ、文学、医術、弁論術、魔術、占星術なども全て
〝Texnai〟のひとつとして考えられていた。



システム開発	■	System Development
システム・エンジニアリング	■	System Engineering
ソフト・ウェア開発	■	Software Development
グラフィック・デザイン	■	Graphic Design
コンピュータ・グラフィックス制作	■	Computer Graphics Laboratory
システム販売	■	System Sales
電子機器輸出入販売	■	International Business

※営業部員、技術者、グラフィック・デザイナー募集。

株式会社テクネ 〒150 東京都渋谷区宇田川町2-1、No. 620 ☎03-464-6927 FAX. 03-476-2372

（資料請求番号 13）

CADの進化論

「三井造船システム技研」が、新しいCADの世界を創造します。

三井造船の長年の実績と優れた技術力を集約して生まれた
「三井造船システム技研」。
CADの設計・製造を行うと同時にCADを使用するという、
ユーザーにしてメーカーの立場から得た総合エンジニアリング・ノウハウで、
つねにCADシステムを進化させていきます。

さまざまな企業・用途に対応する豊富なラインアップ。

設計製図システム

- ●EWS型CAD/CAMシステム
  CADEX®10設計製図システム
- ●ターンキー型大規模CAD/CAMシステム
- ●アプリケーション・ソフトウェア
  機械・電気・板金・金型設計システム SCHEMA
  盤設計製図システム
  射出成形金型システム MOLDA〔3次元ソリッドモデル〕
  配置計画設計システム XCELA〔3次元構造体モデラ〕
  配管計画設計システム XCELA〔3次元構造体モデラ〕
  板金加工システム
  鉄筋加工・積算システム
  指輪彫刻システム MCARES
  図面情報検索システム RAMDA
  NCマシン・サポート・ユーティリティ
- ●グラフィック・ディスプレイ
  7030/7150・20″ラスタ型カラーディスプレイ

地理図形データと非図形データを効率的に入力・編集。

地理情報管理システム

- ●地理情報管理システム U-DAMS
- ●地理データ入力編集システム MERCARTGRAPH
- ●数値写真測量システム MiGEOS

■

CADEX®は三井造船システム技研㈱の総合エンジニアリング・ノウハウを、
コンピュータグラフィックス分野に生かしたCAD/CAM製品グループの総称です。

㈱三井造船システム技研株式会社
CAD/CAM事業部営業部 電話03-544-3455直通
〒104 東京都中央区築地5-6-4

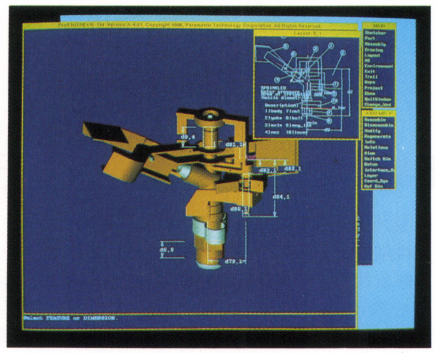
Pro/ENGINEER™

フィーチャー・ベース・モデリング

3次元の概念設計から詳細設計までサポートする画期的なパラメトリック・ソリッド・モデリング・システムです。オブジェクト志向の“フィーチャー・ベース・モデリング”、“パラメトリック”な入力・変更、使い易い“寸法駆動方式”を特長とした、だれでも思いのままに利用できるパラメトリック・ソリッド・モデルです。

- **パラメトリック・スケッチャー機能**
- **パラメトリック・パート・モデラー機能**
- **パラメトリック・アセンブリー機能**
- **パラメトリック・ドローイング機能**
- **パラメトリック・メッシュ機能**
- **カラー・シェーディング、自動高速陰線処理機能**

Pro/ENGINEER™は他のCAE、CAD、CAMシステムとの豊富なインターフェースを用意しており、他システムとの共存利用が可能です。
また、機能別オプション・ソフトウェアの利用により、カスタム化、複雑な3次曲面モデル作成、自動組立、自動メッシュ発生、図面化等、アプリケーションの拡張が可能です。

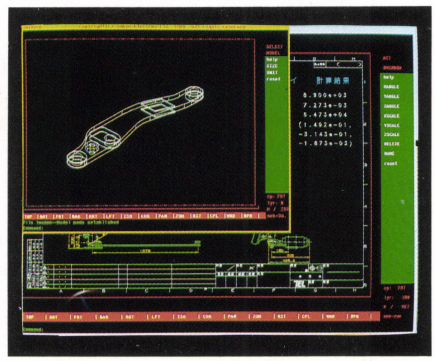


CADWIN™

2D/3D MCAD

CADWIN™は従来型の高機能CADをEWS版にした高機能2D/3D MCADです。CADのコアであるデータ・ベースはサーフェイスまでの豊富な要素タイプをサポートしており、EWSの特長であるマルチ・ウィンドウ機能をふんだんに利用し、その操作性、機能性を飛躍的に向上させました。

- **便利な製図機能**
  JIS寸法線、ワープロウィンドウ機能、ドラフターに近い作図手法
- **高度なCAD機能**
  MODEL/DRAW、CPL/VIEW、シェーディング等
- **拡張性**
  2次元図面～3次元サーフェイスまでの幅広いニーズに対応
- **ユーザー言語**
  LINK-DATによるカスタマイズが可能
- **豊富なインターフェイス**
  従来のCADデータ変換、Pro/ENGINEER™との統合利用



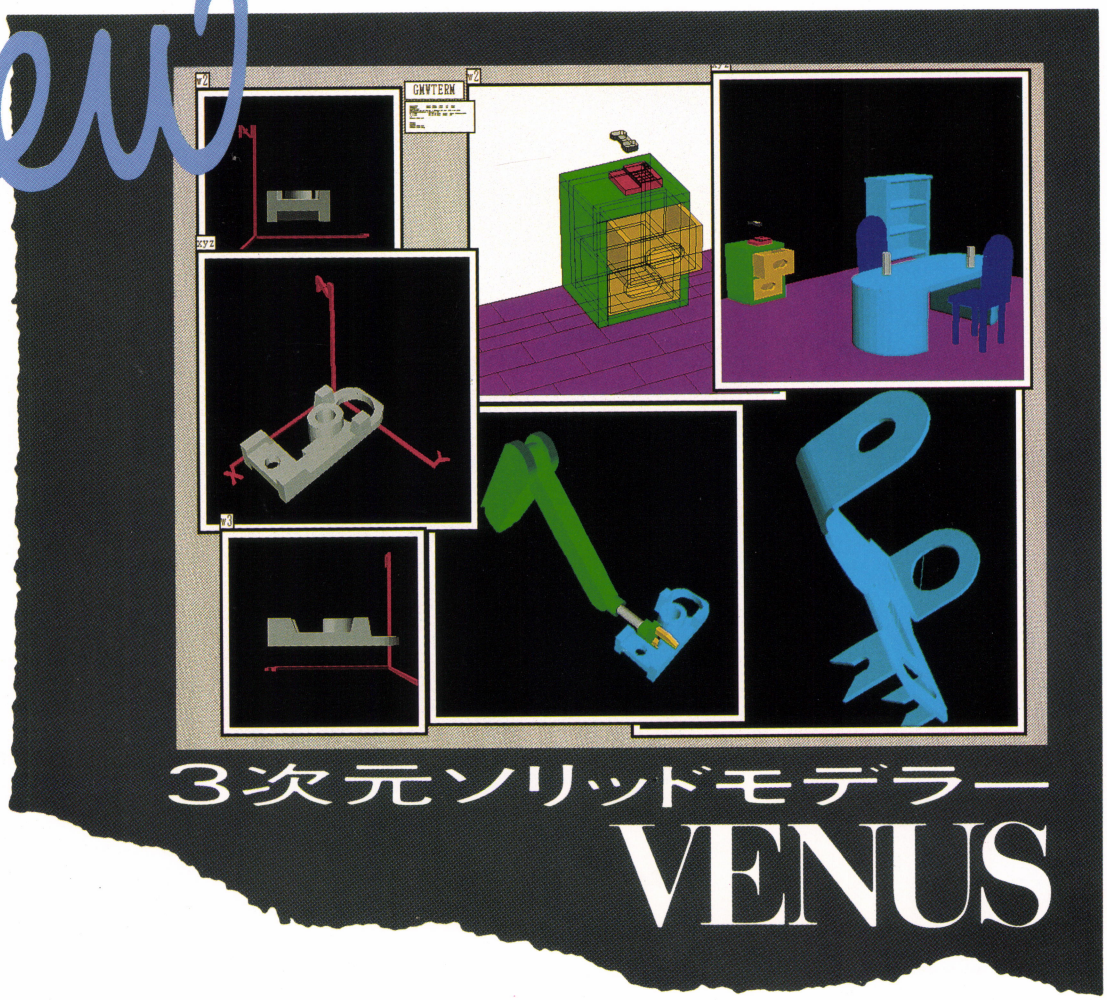
豊かな技術
確かな サービス

東京エレクトロン

■お問合せ・資料請求：東京エレクトロン株式会社
コンピュータ・システム事業部 CIMシステム部
本社：〒163・東京都新宿区西新宿1-26-2 新宿野村ビル ☎03-344-8921

（資料請求番号　15）

New



3次元ソリッドモデラー VENUS

■使い易い■ 3次元ソリッド・モデラーを求めて、次の機能を実現しました。

◉直感的で簡単な操作

対話的にソリッド・モデルの構築が可能です。マウスとエコーによる直感的な形状操作、豊富な局所変形操作により、エンジニアのイメージどうりの立体モデルを表現できます。

●マウス主体の操作 ●3次元ピック機能 ●豊富な局所変形機能 ●変形形状のエコー ●undo機能 ●複数面の同時変形(面セット機能) ●ウィンドウ毎に異なる表示モードで、マルチウィンドウ表示(GMW**) ●複数ウィンドウからの同時アクセス、同時エコー ●日本語処理(Wnn**) ●キーボードからも、マウスからも同等な操作が可能

◉ずば抜けた応答性

従来の同種製品とは、比較にならないほど応答性が向上しました。

●高速演算アルゴリズム ●ブール演算より高速な局所変形操作 ●half edge法によるB-Reps表現 ●C言語による高速処理 ●移植性が良く、常に最新ハードウェアへの対応可能 ●データのメモリ上への常駐による高速化

◉すぐれたコスト・パフォーマンス

32 bit EWS上で稼働、従来システム価格の数分の一を実現しました。

●各社の汎用ワークステーションで使用可能 ●特殊なハードウェア不要
●安価なディスクレスマシンへの対応

●高い改造性

シェル言語(マクロ、制御構造を持つ言語の採用)、画面レイアウト機能(ウィンドウ、メニュー等のレイアウトが自由)により、カスタマイズが容易で、3次元CADのプラット・フォームとして使用できます。

●ネットワーク対応

ネットワーク対応の、マルチ・ウィンドウ・システム(GMW)と、日本語処理システム(Wnn)により、ネットワーク上の複数マシンへの同時表示、CADデータ、ドキュメント類、日本語変換辞書等の集中管理が可能です。

●日本語完全対応

純国産のソフトウェアですから、コマンド、立体名その他への日本語の使用まで考慮されています。

●統合的な環境

マルチ・タスク、マルチ・ウィンドウを利用し、2次元CAD、文書処理システム、構造解析システムとの、統合的な環境の実現を予定しています。

(販売形態) ソース、ロード・モジュールの2種類

*VENUSは情報処理振興事業協会の委託を受け、(株)アステックとデザインオートメーション(株)が、東京大学精密機械工学科 木村研究室の指導のもとに開発したものです。VENUSはVisual Engineering Network Unification Systemの頭文字をとったものです。

**GMW、Wnnは、京都大学数理解析研究所、(株)アステック、立石電機(株)が共同開発したものでフリー・ソフトウェアとして広く配布されているものです。

開発・販売・問い合わせ先

株式会社 アステック
〒150 東京都渋谷区道玄坂1-22-12 長島第一ビル
TEL (03) 477-1541 (代)

デザイン オートメーション株式会社
〒160 東京都新宿区百人町1-13-1 ローズベイビル
TEL (03) 360-8033 (代)

深化 グラフィックス・スーパーコンピュータ。

リアルタイム・グラフィックスの未踏領域を拓く TITAN（タイタン）

コンピュータのニューカテゴリーとして誕生したグラフィックス・スーパーコンピュータ〈TITAN〉。'89年のエンジニアリング環境をさらに深めようとしています。スーパーコンピュータ並の演算パワーと高速3次元グラフィックス機能を、ハイバランスに融合。一挙にリアルタイム・グラフィックスを実現しました。科学技術計算と、その結果をリアルタイムでビジュアライズするリアルタイム・グラフィックスの世界。今日まで、どんなに高価なスーパーコンピュータも、グラフィックス・ターミナルも到達しえなかったこの未踏領域に、〈TITAN〉が第一歩を印したのです。しかも、エンジニア一人一人に手が届く低価格で……。科学技術シミュレーション、モデリングに新次元を拓く〈TITAN〉、クボタコンピュータがお届けします。

Graphics Supercomputer

TITAN

〈構成〉
ハードウェア：本体（1～4プロセッサ並列処理、8～128MBメモリ、380MBディスク3台および120MBテープ内蔵）、19インチカラーモニタ（1670万色）、キーボード、マウス　他
ソフトウェア：UNIX（System V.3およびBSD4.3）、自動ベクトル化自動パラレル化コンパイラ（FORTRAN-C）、VAX/VMS互換コンパイラ（FORTRAN）、NFS、Doréグラフィックツール、TCP/IP、イーサネット、X-WINDOW、日本語機能　他
〈性能〉
ピーク演算性能：16～64MIPS、16～64MFLOPS（1～4プロセッサ）
倍精度 LINPACK：6.1MFLOPS（1プロセッサ時）、10MFLOPS（2プロセッサ時）
グラフィック性能：5千万グローシェーディドピクセル/秒、40万ショートベクトル/秒、7.5～20万ポリゴン/秒（1～4プロセッサ）

＊UNIXはAT＆T社の開発したOSです。
＊VAX/VMSはDIGITAL EQUIPMENT社の登録商標です。
＊NFSはSun Microsystems社が開発しライセンスしているソフトウェアです。
＊イーサネットはXEROX社の登録商標です。

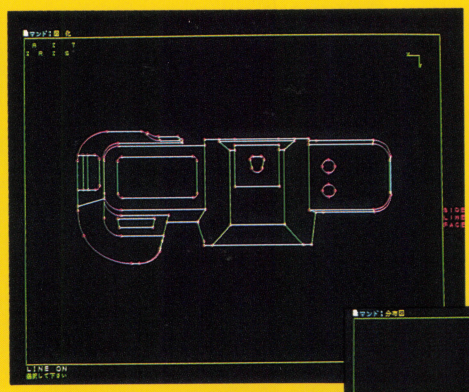
クボタコンピュータ株式会社

〒160 東京都新宿区新宿2-8-8 ☎03(225)0741　大阪支店 〒541 大阪市東区本町5-15-2 ☎06(264)2501　山梨工場 〒400-02 山梨県中巨摩郡白根町下今諏訪907-8 ☎0552(84)4861

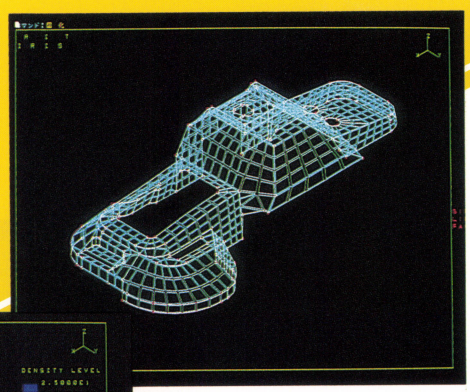
操作性を重視した対話型統合CAEシステム

IRIS

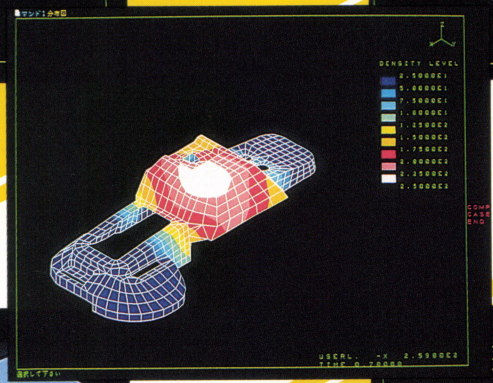
IRISはエンジニアリング業務の効率化を図るため、設計から解析・評価さらには製図、技術レポート作成と幅広くサポートするソフトウエアパッケージ群です。IRISは、対話形式でポップアップ・メニューをマウスあるいはペンで指示することで必要な機能を操作でき、また形状の生成においても、ディスプレイ上に表示されている形状あるいはコマンドを指示することでシステムを操作できます。



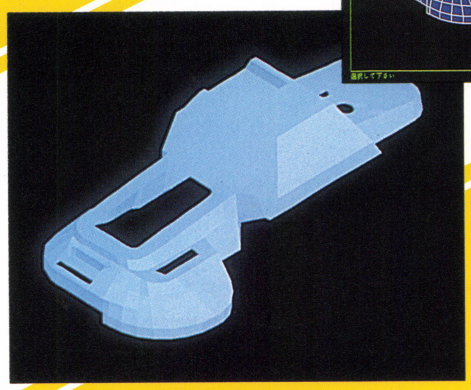
DRAFT



G-MESH

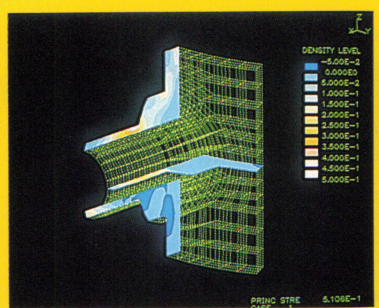


R-VIEW

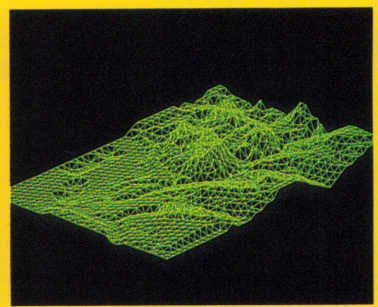


DESIGN-3D

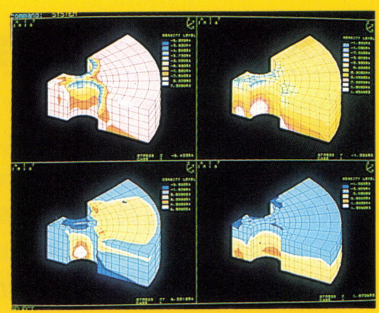
COMPOSE



R-VIEW



G-MESH



R-VIEW

株式会社 エイ・アイ・テクノロジー

〒102 東京都千代田区東神田2-6-2 タカラビル9F
TEL 03-864-6912㈹

男女ソフトウエアエンジニア募集中

コンピュータグラフィックスに興味のある方は履歴書を送付、またはご連絡下さい。 担当 広瀬

■販売代理店

センチュリリサーチセンタ㈱ 営業開発第1部 〒103東京都中央区日本橋本町3-6-2 TEL 03-665-9741
ニチメンデータシステム㈱ 営業部 〒103東京都中央区日本橋本町1-3-2共同ビル TEL 03-241-2611
メモレックス・テレックス㈱ EWS営業部 〒103東京都中央区八重洲1-7-20 八重洲口会館 TEL 03-275-0452
東芝エンジニアリング㈱ EWSシステム部 〒210 神奈川県川崎市幸区堀川町66-2 興和川崎西口ビル TEL 044-548-3354
日本鋼管㈱ コンピューター・機器営業部 〒102 東京都千代田区九段北4-1-3 飛栄九段北ビル TEL 03-222-5965
キヤノン販売㈱ 特機営業部 〒108東京都港区港南2-13-29キヤノン・オプティカルセンター TEL 03-740-3334
日本電子計算㈱ 科学技術事業部 〒103東京都中央区日本橋兜町6-7 TEL 03-668-6171

JRC 日本無線

MAPPING GRAPHICS 450

SpaceGraph 450は、ナンバーワンの実績を誇る410シリーズの機能をさらに革新させ、600,000ショートベクトル/秒の描画性能を実現したマッピング専用のグラフィックディスプレイです。その超高速表示性能と比類なき応答性は、地図の緊急検索システムや大量データを持つ新たな地図情報処理システムの実現を約束します。

SUPER GRAPHICS 250

- 専用のカスタムLSIで超高性能化を実現
- 600,000ベクトル/秒以上の超高速表示
- 高度なインタラクティブ機能
- 多彩なインタフェースを装備
- 豊富な入出力装置
- 実績のあるグラフィックパッケージ

JRC 日本無線株式會社

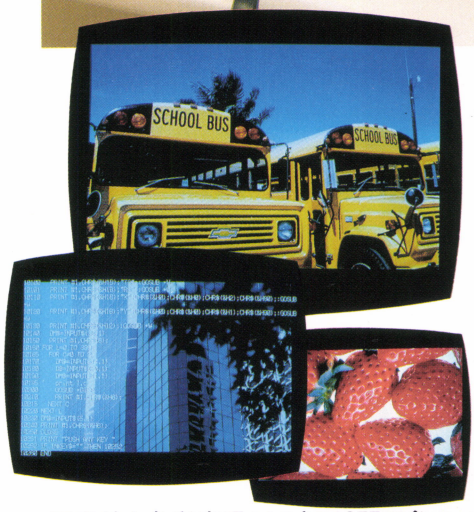
お問合わせは電子営業部情報機器課へ
本社事務所　〒107 東京都港区赤坂2丁目17番22号　赤坂ツインタワー本館
☎ (03)584－8821(ダイヤルイン)　ファクシミリ(03)584－2482

●北海道支社(札幌)／☎(011) 261－8321
●青森支店(八戸)／☎(0178) 33－5222
●四国支店(高松)／☎(0878) 33－9800
●東北支社(仙台)／☎(022) 225－6831
●東海支店(清水)／☎(0543) 53－0138
●中国支店(広島)／☎(082) 243－0686
●関西支社(大阪)／☎(06) 344－1640
●中部支店(名古屋)／☎(052) 203－1225
●長崎支店／☎(0958) 61－8148
●九州支社(福岡)／☎(092) 761－2636
●神戸支店／☎(078) 321－2431
●鹿児島支店／☎(0992) 23－5261

C&C Computers and Communications

NEC

自然の色を表現する1677万色入力。



■微妙な色彩表現ができる1677万色のカラー入力。色彩豊かなイラストや微妙な色調の写真などのカラー原稿を、1画素当たりRGB各最大8ビット、256階調の高精細で読み取り、1677万色のフルカラーを再現します。
（データ保証値は、64階調・26万色）

フルカラー、高速・高精細読み取りで、あらゆるグラフィック・ニーズに応えます。

NEW

■多彩な機能が鮮明で自然な表現を可能にします。高性能ディザ機能により、凝似的に自然画に近い表現ができます。また、輪郭強調機能を搭載し、文字と図形の重なった部分でも文字をはっきりと読み込むことが可能です。

■最大A4サイズまでの読み取りが可能。ビジネスの高度なニーズにも余裕の読み取りサイズで応えます。

■線密度等の設定状態が一目でわかる液晶ディスプレイを装備。複雑な読み取り条件をわかりやすく表示し、快適な入力を実現します。

■接続を容易にする。豊富なインタフェースを用意。標準装備のRS-232Cインタフェースに加え、オプションでGP-IB/パラレルインタフェースを用意。

NECパーソナルコンピュータ

カラーイメージスキャナ PC-IN506

本体標準価格 228,000円

あらゆるジャンルで手軽に使える、A6サイズ対応のパーソナルカラーイメージスキャナ。

NEW

PC-IN505
本体標準価格138,000円

※記事中の使用例は、原稿をイメージスキャナで読み込んでソフトウェア処理したものです

NEW

グラフィックス自由自在
微妙な濃淡もくっきり。どんな画像も
鮮やかに読み取る64階調の高精細、多機能イメージスキャナ。
イメージスキャナ PC-IN503G　本体標準価格145,000円

■パーソナルイメージスキャナPC-IN502 本体標準価格99,800円
多彩な画像データに対応するスタンダード機。
■ハンディイメージスキャナPC-IN511 本体標準価格69,800円
A6サイズまでの読み取り可能。手軽に使えるハンディタイプ。

日本電気グループ〈技術的なご質問・ご相談に電話でお答えします〉
NECパソコンインフォメーションセンター
受付時間…9:00～17:00　月曜日～金曜日（祝日を除く）
東京03(452)8000　大阪06(211)9800 電話番号は、よくお確かめのうえおかけください。

（資料請求番号　20）

TOSHIBA

個性を伸ばすと、会社も伸びる。

技術者の個性と創造力を伸ばすには、時間のゆとりも欠かせません。
導入しやすく、拡張性に富んだCS-CADDSネットワークは、
技術者1人に1台のゆとりを実現。仕事の質と一緒に、会社も伸ばします。

東芝の優れたEWS技術と世界中で豊富な実績を持つCADDSソフトウエアが融合しました。3次元モデリング・解析など高度な作業を支援するCADDStationと手軽に2次元設計・製図を行うパーソナルCAD CS-3100/HR。この2つのシステムを結んだCS-CADDSネットワーク。技術者の個性と創造性を確実にシステムに活かせます。

■CAD間のデータを共有

CADDStationとCS-3100/HRで、双方向のデータ交換が可能です。このCS-CADDSネットワークには、転送速度10Mbpsという高速のイーサネット，TCP/IPを採用。大量データの転送もスピーディにできます。

■資源の共有化

プリンタ、プロッタ、及びハードディスク等の資源を共有。ネットワーク内で効率のよいシステム構築ができます。

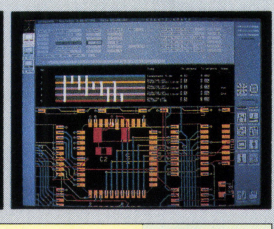
■機能分散

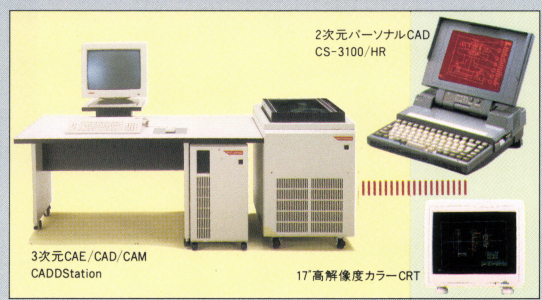
業務の形態に応じ2次元作業、3次元作業の分担ができます。個々のシステムをフル活用する組み合わせをフレキシブルに構成できます。

■CPU分散

コンピュータ資産を活かしながら、必要に応じてシステム拡張ができるAdd-on方式です。

※CADDS：Computervision Automated Design and Drafting System
※CADDSはCOMPUTERVISION社の開発ソフトです。





東芝CAE/CAD/CAM
CADDStation/CSシリーズ

お問合せは 株式会社 東芝/ コンピュータ事業部 電話(03)457-2725(ダイヤルイン)
産業システム事業部 電話(03)457-4340(ダイヤルイン)
〒105 東京都港区芝浦1-1-1(東芝ビルディング)

先端技術をオフィスに…エネルギーとエレクトロニクス E&Eの東芝

（資料請求番号 21）

イメージの世界が翼をひろげます。

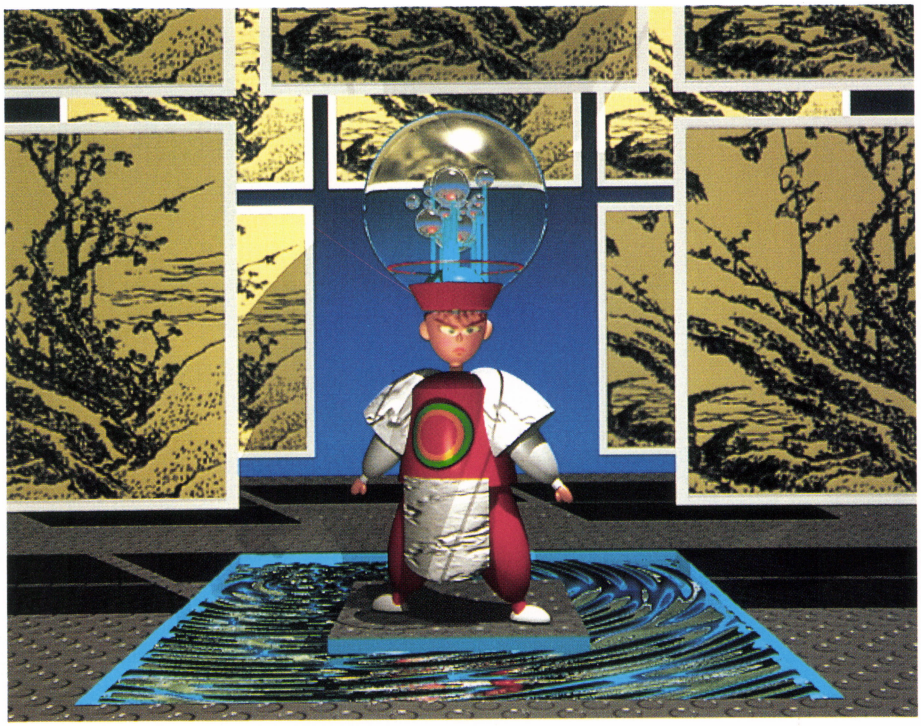
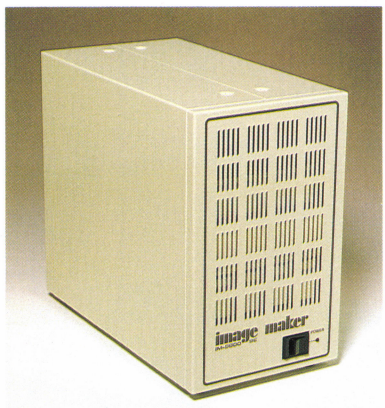


image maker
IM-9800

●2Dペイント ●3Dアニメーション ●IM RAY-TREKⅡ ●3DM ●DTP4U

新製品

- ●並列処理でCG画像制作の大幅な時間短縮を可能にする **SUPER ENGINE IM-9800SE**
- ●IM-9800SE対応レイトレーシングソフト **IM SUPER-TREK**
- ●高品位の建築プレゼンテーションを可能にする **IM-DYNAPERS**



YDK

株式会社 ワイ・デー・ケー

新宿営業本部 TEL 03(342)2471
〒163 東京都新宿区西新宿2-4-1 新宿NSビル1F
大阪支店 TEL 06(532)7051
〒500 大阪市西区阿波座2-2-1 サンキンビル内

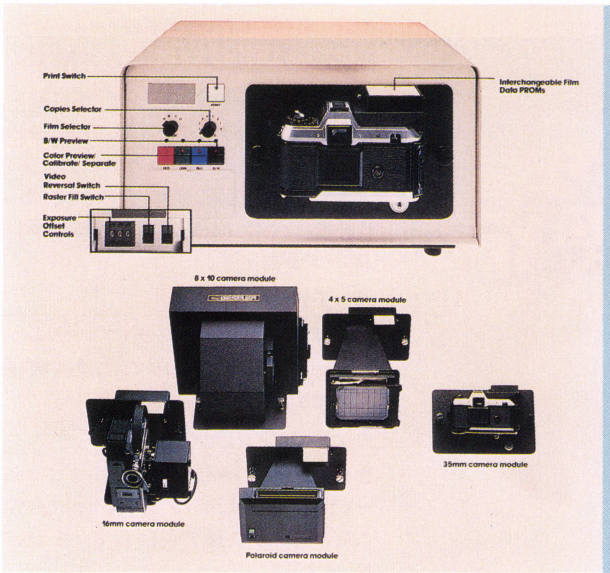
（資料請求番号 22）

TRUVEL

立体物スキャナ

カラー／カラー、モノクロ／モノクロ用の3機種

- ●最高50㎜、最大300×430㎜までの立体、平面被写体を
- ●スキャニングスピード、B5サイズの原稿で、最高モノクロ8秒、カラー42秒
- ●高解像度、300dpi～900dpi　●SCSIインターフェース
- ●24bit、CCD 1パススキャン方式



//ICHIKIS

2000×1360分解能

ノンインターレス／マルチスキャン方式／フィルムレコーダ

- ●最大150MHz　●0.08㎜スポットサイズ
- ●カットグラスCRT　●偏向角70°
- ●ステータ型偏向ヨークコイル　●超微粒子蛍光体
- ●豊富なカメラモジュール（16㎜、35㎜、4×5、8×10、ポラロイド）

Pixelogic

24ビット

カラー ビデオ デジタイザ

- ●R.G.B.、NTSCおよびVHSビデオ入力
- ●480×640画素　●SCSIインターフェース
- ●最大1670万色または256グレイスケール

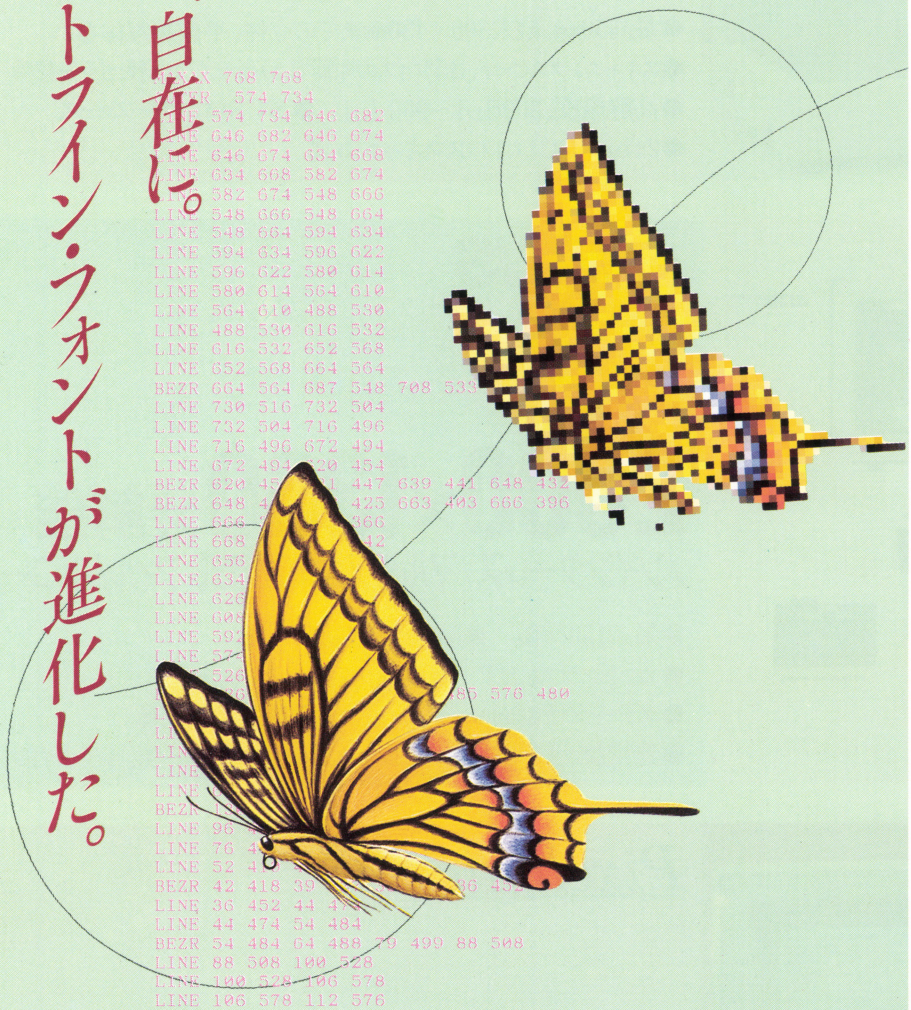
※処理速度1/30秒（16グレイスケール時）

日本バイナリー株式会社
〒150 東京都渋谷区渋谷2-17-3南塚ビル
TEL.(03)407-9751(代表)

（資料請求番号　23）

真mana名

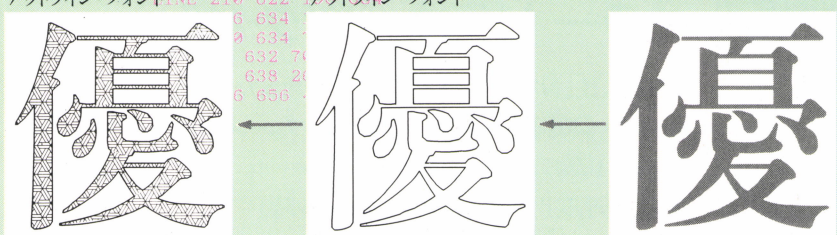
美しく、自在に。
アウトライン・フォントが進化した。



ハッチング処理をした
アウトライン・フォント

作成された
アウトライン・フォント

文字元パターン



高品位フォントをパソコン上で作成する「真名」、日本語DTP時代のニーズに応えて、新登場。

パソコン上で稼働

真名はパソコン上でアウトライン・フォント作成を実現します。特別なハードウェアは必要なく、従来のようにミニコンやEWSを使用する場合に比べ、低コストです。

ドット・パターン編集機能

スキャナー、TVカメラよりドット・パターンを入力できます。マウスにより読み込んだパターンに追加、修正が可能で、スムーズ拡大機能や入力ノイズの修正機能など高度な編集機能も装備しています。

円弧、ベツェール曲線のあてはめ

真名では、特殊なアルゴリズムの開発により、曲線部の円弧、ベツェール曲線のあてはめを高速に自動的に行います。ドット・パターンでは不可能な美しいアウトライン抽出を実現しました。

アウトライン編集機能

作成されたアウトラインは、直線、円弧、ベツェール曲線として修正、追加が可能です。文字の大きさの自由な選定、回転、白抜きなども容易に実現。パソコンCADクラスの編集機能を持っています。

バッチ処理機能

ユーザーが設定したバッチ・プログラムにより、同一処理の多いアウトラインの抽出を自動実行することが可能です。

カッティングプロッターに対応

作成されたアウトラインは、カッティングプロッターで出力できます。ポップ文字やロゴタイプの作成はもちろん、各種デザイン作業に威力を発揮します。

新発売

真mana名

アウトライン・フォント作成システム
NEC PC-9800シリーズ用

価格 150万円(MS-DOSは別売)

使用条件

- **パソコン**:NEC PC-98XA、PC-98XL、PC-98XL²
  入力装置:NEC PC-IN502およびTVカメラ等
  CRT:ハイリゾリューションディスプレイ
- **メモリ** 768Kバイト実装　**MS-DOS** 3.0以上
- マウスが必要です。ただし、アウトラインの抽出部分のプログラムには必要ありません。
- アウトライン・フォントの編集プログラムについては80287等のNDPが必要です。

※MS-DOSは米国マイクロソフト社の登録商標です。

総発売元
Mitsubishi Corporation
三菱商事株式会社
取扱セクション
技術部インフォーメーション テクノロジーチーム
〒100-86 東京都千代田区丸の内2丁目6番3号　TEL.03(210)2538、7543、7386

開発元
ICL
Information and Control Laboratory Co., Ltd.
株式会社インフォメーション アンド コントロール研究所
〒160 東京都新宿区新宿5丁目11番22号中島ビル5F　TEL.03(352)4746 FAX.03(357)7114

情報は友だち
RICOH

A3判出力のレーザーだから、

オフィスで使える。設計分野で使える。
B4・A4判への縮小印字も自由自在。
さらに、各社のパソコンに接続できる。
高性能レーザープリンター、
“PC LASER”ついに誕生。

**●エミュレーションカードで
さまざまなパソコンに対応。**

エミュレーションカード(ICカード)を差しかえるだけで、各社のパソコン用プリンターとして、威力を発揮します。
適応機種:NEC PC-9800/N5200、TOSHIBA J3100、日立 B16、リコー Mr.マイツール

●お手持ちのソフトの変更は不要。

現在お使いのドットプリンターとプラグコンパチブル。適応機種に対応するほとんどの市販ソフトは、そのまま使えます。

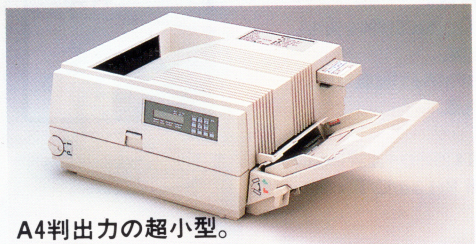
●A3判原稿のB4・A4縮小印字も自由自在。

プリンター上で設定できる縮小サイズは5種類。操作は、パネルの縮小キーを押すだけなのでとても簡単。ご希望のファイリングサイズで、鮮明なプリントアウトが行えます。

●フォントカートリッジひとつで対応できる豊富な文字フォント。

LP3320-SP4

標準システム価格……628,000円
(LP3320-SP4、エミュレーションカード
フォントカートリッジF24M 含む)



A4判出力の超小型。
姉妹機、LP1060-SP3も好評発売中。

パソコン用レーザープリンター
PC LASER

●カタログを差し上げます。ハガキに資料請求券を貼って右記住所までお気軽にお送りください。〒107 東京都港区南青山1-15-5 株式会社リコー宣伝部 リコー
また、電話によるお問い合わせは☎03(578)3111 I/O営業部計画課までお気軽にどうぞ。

資料請求券
LP3320-SP4
PIXEL
1/20号

NISA Ⅱ/DISPLAY Ⅱ on Sony News

特別価格キャンペーン実施中!!

1989年3月31日まで

汎用構造解析ソフト
NISA Ⅱ/DISPLAY Ⅱ
世界2,500社で使用されている、
構造解析ソフトウェア。

＋

Sony News
NWS-841
キーボード、マウス
17″高解像度カラーモニター
カラービットマップボード付

＝ キャンペーン価格 **625万円**

NISA Ⅱ

NISA Ⅱには以下の解析機能が含まれます。

- ●線形静解析
  - －複数荷重ケースとその重ね合わせ
  - －等価応力計算
- ●非線形静解析
  - －超弾性材料(ゴムなど)
  - －温度依存性材料
  - －弾塑性解析
  - －大変形、大回転、有限歪み
  - －応力剛性
- ●熱伝導解析
  - －線形、非線形解析
  - －定常、非定常解析
  - －対流、伝導、放射
  - －相変化
- ●動解析
  - －固有値解析
  - －過渡応答解析
  - －周波数応答解析
  - －ショックスペクトル解析
  - －ランダム振動解析

DISPLAY Ⅱ

- ●プリプロセッサ
  - －3次元対話型カラーグラフィクス
  - －3次元形状モデリング
  - －モデルチェック機能
- ●ポストプロセッサ
  - －各種の変形図、応力図
  - －特定点の時間履歴

NISA Ⅱの特長

- ●NISA Ⅱファミリーは広範囲の解析機能を提供する総合CAEソフトです。
- ●NISA Ⅱファミリーは高機能、低価格です。
- ●NISA Ⅱファミリーはパソコンからスーパーコンピュータまで同一のソフト体系です。
- ●NISA Ⅱファミリーは各種CAD/CAMシステムに幅広く対応しています。

別売ソフト

- ●最適設計(NISAOPT)
  - －構造の最適化
  - －形状の最適化
  - －梁構造物の最適化
- ●流体解析(NISA3D-FLUID)
  - －3次元流体解析
  - －圧縮流/非圧縮流
  - －定常/非定常流
  - －層流/乱流
  - －相変化を伴う流れ
  - －ボーラスメディア
- ●疲労/破壊解析(ENDURE)
  - －クラック発生予測
  - －クラック進展予測
  - －静解析/動解析との直接インターフェイス
- ●プリント基板解析(FEAP)
  - －PCBの熱、応力、振動の解析
  - －電子部品の形状ライブラリーを完備
  - －自動メッシュ生成

＊NISA Ⅱファミリーは、EMRCが開発しライセンスしています。

お問い合わせ先

- ●センチュリリサーチセンタ株式会社
  〒103 東京都中央区日本橋本町3-6-2 Tel.03(665)9828
- ●ヂーゼル機器株式会社
  〒150 東京都渋谷区渋谷3-6-7 Tel.03(5485)4160
- ●久保田鉄工株式会社
  〒103 東京都中央区日本橋室町3-1-3 Tel.03(245)3836
- ●メモレックス・テレックス株式会社
  〒103 東京都中央区八重洲1-7-20 Tel.03(275)0452

SONY

〈NWS-841の主な特長〉

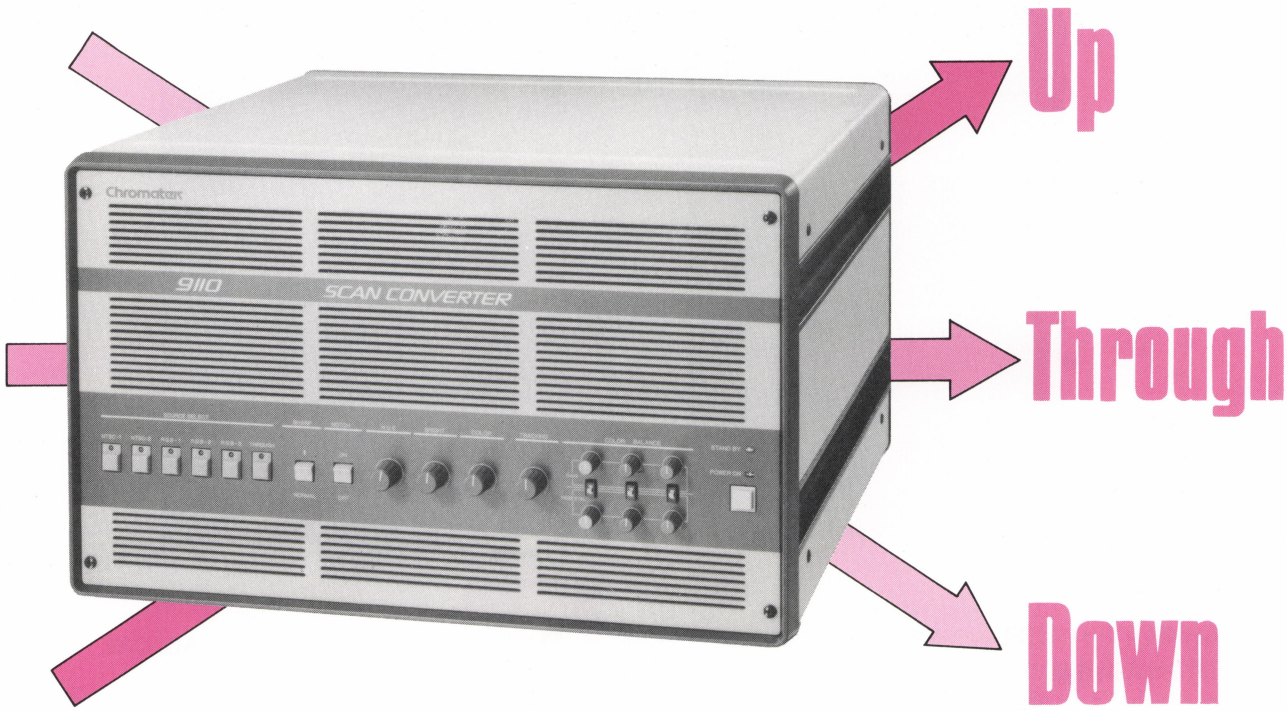
- ●CPU、I/Oプロセッサともに、MC68020を使用し処理速度2.4MIPSを実現。
- ●メインメモリ8MB、最大16MBまで拡張可能。
- ●286MBハードディスク内蔵。
- ●標準ソフトウェア
  NEWS-OS2.2：UNIX4.2BSD＋X Window System Ver 10＋NFS Rel 3.0/C, Fortran, Pascal etc.

開放型分散処理ワークステーション
NEWS

※UNIXは、AT&Tのベル研究所が開発し、AT&Tがライセンスしています。
※NFSはSun Microsystems社が開発し、ライセンスしています。

9110
UNIVERSAL SCAN CONVERTER
高速リアルタイム・ユニバーサル・スキャンコンバータ

ビデオタイミングのアップ、ダウンが思いのまま。



あらゆる映像をVTR収録

高解像度CAD/CAM画像，リモートセンシング画像，各種PC/EWS画像，ハイビジョン映像，各種ニューメディア画像をリアルタイムでNTSCタイミングに変換しプレゼンテーションや放送・ビデオパッケージの制作に利用できます。

限られたディスプレイを有効利用

1台しかないビデオプロジェクタやCRTにあらゆる映像ソースが表示できればプレゼンテーション効果はもっと上がるはずです。またCRTディスプレイはたくさんあるのに表示タイミングが異なる、ビデオプリンタにも出力したいがタイミングが合わないなど出力側が限定されていても、あらゆる入力タイミングを任意の出力タイミングに「9110型」が変換します。

■主な仕様

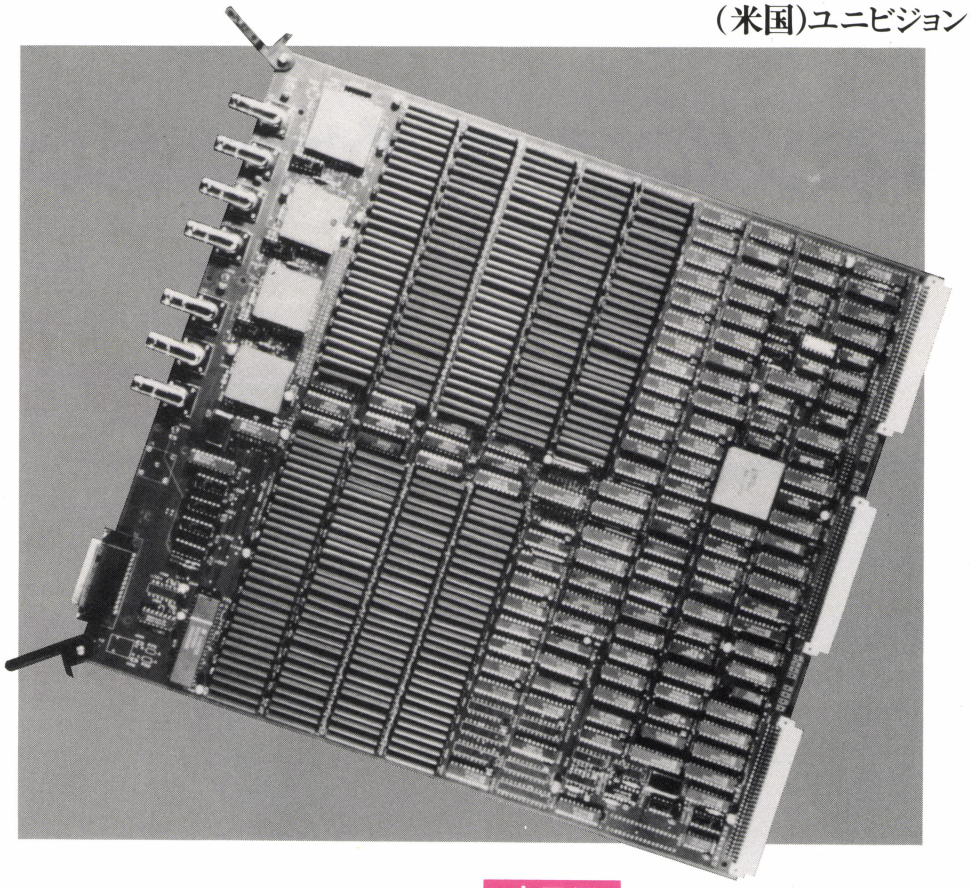
- ●水平同期周波数：15KHz～64KHz
- ●垂直同期周波数：25Hz～240Hz
- ●インターレース比：1対1または1対2
- ●標本化周波数：入力側120MHzMAX
  出力側125MHzMAX
- ●最大画面構成：2032(H)×1016(V)
- ●最大変換速度：55MPIXEL/sec
- ●入出力同期タイミング：入出力独立ユーザプログラマブル
- ●入力信号：NTSC・RGB・デジタルRGB(スイッチ切替え)
- ●電源：AC-100V±10% 消費電力350VA
- ●外形寸法：471(W)×280(H)×462(D)mm

ASTRODESIGN,Inc.
アストロデザイン株式会社　神奈川県川崎市中原区上小田中863 〒211 TEL(044)751-1011 FAX(044)751-1300
関西営業所　大阪市東淀川区東中島1-18-27新大阪丸ビル新館510 〒533 TEL(06)328-8558 FAX(06)328-5058
製造開発：クロマテック株式会社

Sun-3/4用フルカラーグラフィックボード

UDC-3400 Series

(米国)ユニビジョンテクノロジー社



概 要

UDC-3400シリーズは、SUNワークステーションのバスフォーマットで設計された高性能フルカラーグラフィックボードです。イメージメモリは2048×1024dotの場合、(24bit)+(8bit Z Buffer)+(2bit overlay)の構成まで可能で、ディスプレイコントローラ82786による高速描画が行えます。

応用例

- ■ソリッドモデリング
- ■CAD/CAE
- ■医療画像処理
- ■アニメーション
- ■リモートセンシング画像の解析
- ■各種プレゼンテーション用イメージ生成

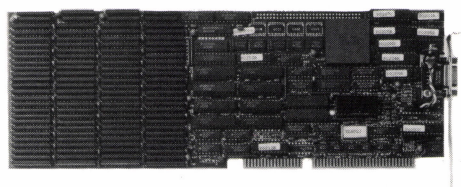
仕 様

- ●高性能グラフィックプロセッサ Intel82786を使用
- ●最大1280×1024の表示解像度(60Hzノンインタレース)
- ●最大2048×2048の表示メモリ
- ●ビットマップメモリ　：8Mbytes
- ●ディスプレイリストメモリ　：$\frac{1}{2}$Mbytes
- ●同時表示カラー数　：1677万色
- ●ビデオバンド幅　：107MHz
- ●ビデオ出力　：RS-343
- ●対応モニタ　：64KHz(水平走査)
  　：60Hz(ノンインタレース)
- ●VMEサンマイクロシステムフォーマット準拠
- ●メモリアクセス　：バイト/ワード/ロングワードアクセス
  - ・ワードアクセス時　：ホストバスより2～400万ピクセル/秒
  - ・ロングワードアクセス時　：VSBバスより1000万ピクセル/秒
- ●レジスタアクセス/メモリアドレッシング　：メモリマップド方式

●Sun-3は米国サンマイクロシステムズの製品です。

UDC-800/2600 Series

UDC-800及びUDC-2600シリーズはIBM-PC/AT又は、互換機組み込みタイプのグラフィックボードです。最大1600×1280の超高解像度で256色又は、1600万色の同時表示が可能です。

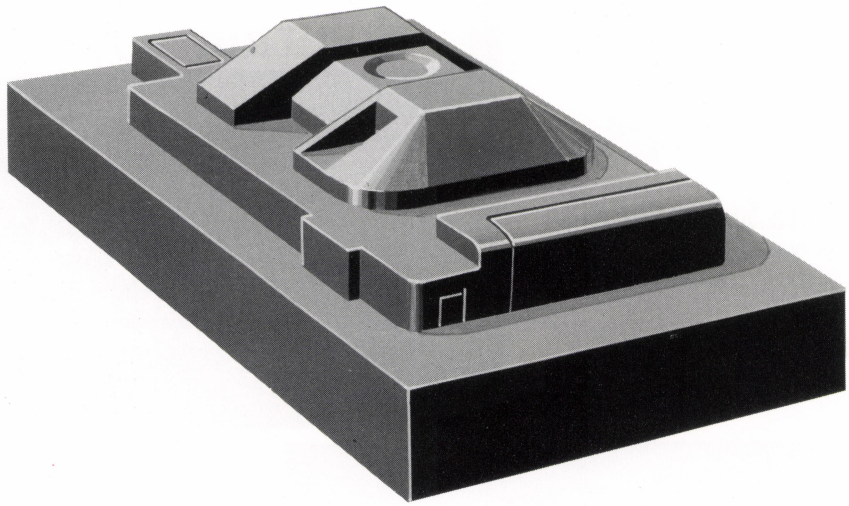


ASTRODESIGN,Inc.

アストロデザイン株式会社　神奈川県川崎市中原区上小田中863 〒211 TEL(044)751-1011 FAX(044)751-1300
関西営業所　大阪市東淀川区東中島1-18-27新大阪丸ビル新館510 〒533 TEL(06)328-8558 FAX(06)328-5058

KOMATSU
Three-Dimension Plotter

設計者のイメージをすばやく立体モデルに具現化するコマツ3次元プロッタに、従来機をさらに上回る高性能機が誕生しました。19.2Kボーの高速データ伝送と高性能CPUのマルチ制御により、最高切削速度は毎分7.5mで、高精度加工を実現。設計・試作期間の大幅な短縮に、また、開発業務の質の向上に貢献致します。



高精度/最高切削速度＝毎分7.5m

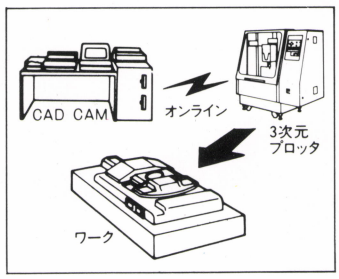
高速3次元加工ならコマツです。

主な特長

▶ワークサイズ：400×315×225mmと拡大しました。
▶スケール機能により、大型モデルの縮小切削、小物の拡大切削が可能です。
▶設置面積は従来機比25%削減。より一層の機動性と省スペースを実現しました。
▶高性能主軸モータ採用：ACスピンドルモータ 1Kw 2,000～16,000rpm
▶高精度加工を達成：精度0.03mm 繰返し精度0.01mm以下
▶ワックス、アクリル・ABS樹脂等プラスチック、木材など各種素材が使用できます。

幅広い用途に

●あらゆるタイプのCAD CAMシステム・自動プログラミング装置の端末機として、設計段階での形状確認に。
●各種型の形状確認や工具干渉などのチェックに。
●プラスチック部品の高速加工や、彫刻、木工作業などへ応用し、多品種短納期に対応。
●ならい機械用のモデル製作に。
●簡易型の製作に。



コマツ3次元プロッタ

小松製作所
電子機器製造部
営業課
〒254 神奈川県平塚市四の宮2597
TEL.(0463)23-1313㈹ (0463)23-5250(直通)
FAX.(0463)23-2917

HOOPSバージョン日米で同時に発売開始
ワークステーション用
基本パッケージ価格、120万円!

完全三次元 CAD/CAM ソフトウェア

Solution 3000シリーズ

「SOLUTION3000」は設計・製造の現場から生まれ、大型システムを超える本格的なCAD/CAMソフトウェアとして、その応答性、データ容量、機能などに加え、基本パッケージで作成された三次元データが、そのままNCへのデータとして利用できる「ALL-IN-ONE」設計が高い評価を受け、既にデトロイト周辺のGMやフォード、クライスラーなどで急速に採用されはじめています。しかも今回発売される新バージョンではベースになるグラフィックス・ライブラリィとして注目を集めている「HOOPS™」を採用することにより、高度な三次元表現を実現すると同時に、HOOPSが稼動している多くの機種で利用できる統合CAD/CAMソフトウェアとしてパワーアップされました。

SOLUTION3000™

完全三次元のデータベースを持つCADソフトウェア
●豊富な作図/編集機能 ●強力な投影機能による三次元ワイヤー・フレームモデル ●どのような三次元空間にある図形も任意の角度から表示できるマルチビュー ●ユーザが簡単に専用化することができるオープンアーキテクチャ スクリーンメニュー/プルダウンメニュー/ポップアップメニュー/アイコンメニュー/タブレットメニュー/ショートキー/マクロ/ビュー他 ●他のCADシステムとのデータ交換 DXFサポート/各自動車メーカーの標準フォーマットにも対応

〔Designer Surfaces〕

●4＊4のコントロールポイント ●Bezier/U&Vサーフェース
●Revolution ●Ruled ●Swept ●Blended Swept ●Cone
●Spheres ●Cylieders ●Toroids ●Blended Fillets
SOLUTION3000で作成されたデータをそのまま利用できます。

〔DGL (Designer Graphics Language)〕

SOLUTION3000の機能を専用化するためのプログラミング言語
●データベース中の情報の管理やループ、変数設定、演算、ファイル管理などを行えます。

〔Advanced Coons-Patch Surface Design〕

メインフレーム・クラスのCOONS-SURFACE
DESIGNER SURFACESで作成されたデータは簡単な変換により利用できます。 IGESサポート

〔N/C Programing Software〕

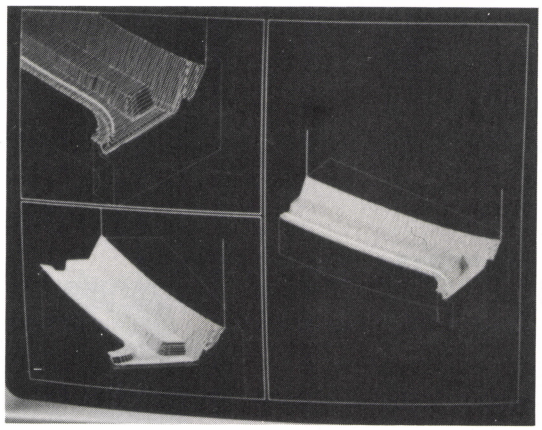
フラット/スカルプチュア・マシニング
インサイドコーナー・カット
2軸から5軸までサポート ドリル/ミリング
DGLによりあらゆるTool Pathに適用可能
ゼネラルポスト・プロセッサ
CLデータ出力

〔Millmaster™〕

(Multiple Surface Machings)
●マルチ・サーフェースを持つカッター・パスのバッジ処理を行うことができます。

〔Digimaster〕

●デジタイズによる2次元図面の3次元入力ソフトウェア



ロボットシュミレーション・ソフトウェア

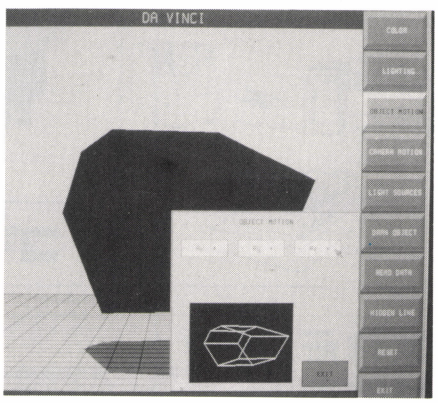
IGRIP™

●各自動車メーカー、航空機メーカー、NASAなどで採用されている高度なリアルタイム・シュミレーションソフトウェア
●対話形式のメニューや専用の簡易言語によってロボット・シュミレーション以外のFA関係のシュミレーションなどにも簡単に応用できます。

オブジェクト指向の三次元グラフィックス・ライブラリィ

HOOPS™

●従来までの手続き型ライブラリィと異なり、簡単な宣言によって高度な三次元グラフィックス表現が行えます。
●HOOPSをベースにして作成されたアプリケーションは、HOOPSがサポートする汎用機からパソコンまでの機種で共通に利用できます。
●主な機能にはウィンドウ機能、図形変換、隠線/隠面消去、光源設定、カメラ機能、入力マネージャ機能などがあります。
●CやFORTRANなどで作成されたアプリケーションからコールすることも可能です。



SOLUTION 3000™/Millmater™はMicro Enginering Solutions社、HOOPS™はIthaca Software社の、IGRIP™はDeneb Robotics社の登録商標です。

FAMOTIK
ファモティク株式会社
〒150 東京都渋谷区代官山町7番8号
ドッケン代官山ハイツ404号
TEL (03)780-4681 FAX (03)780-4682

マドレールSF

Ver.2.0

パソコンでフルカラーのソリッドモデリング

『マドレールSF』は、円柱、円錐、球、回転体、柱体といったプリミティブのブーリアン演算を行うことにより、複雑な3次元物体を生成するソリッド・モデラーです。モデリングはキーボードまたはプログラムによってインターラクティブに行えます。自動的にプログラムを生成するレコード機能も備えています。

データ表現はB-reps(境界表現)となっています。入力および出力部分は公開されていますので、他のアプリケーションのCSG (Constructive Solid Geometry)形式のデータを『マドレールSF』の入力データとしたり、『マドレールSF』でモデリングしたものを他のワークステーションやミニコンなどでレンダリングしたりすることが可能です。

稜線表示、簡易陰線消去表示、陰線消去表示、半透明なコンスタント・シェーディング表示、レイトレーシング表示といった表示モードがあり、これらのモードを組み合わせた複合表示も可能です。フレームバッファを使用すれば、1677万色を使ったシェーディング表示が行えます。半透明なコンスタント・シェーディング表示は稜線表示と同程度の速度で表示をすることができます。

数値演算プロセッサ8087、80287、80387を使用すれば高速な演算が可能となります。数値演算プロセッサがなくとも高速な演算が可能なプログラムも用意されており、ラップトップ・パソコン上でモデリングすることも可能です。

現在対応しているパソコンは、PC−9801およびPC−286シリーズです。フレームバッファはサピエンス社のスーパーフレームが使用可能です。フレームバッファが実装されていない場合はPC−9801のグラフィックス画面に4096色相当のタイリング表示で出力します。

※東京秋葉原T-ZONEにて『マドレールSF』のデモを行っております。
※『マドレールSF』、『UCG』ともにPC-9801RA、MS-DOSバージョン3.3に対応しています。

●価格／¥118,000
●開発・販売／株式会社　アークブレイン

浅葉肇　作

鈴木馨　作

Ultra C Graph Ver.1.52

PC-9801、PC-286対応 C言語用超高速 グラフィックス・ライブラリ

■Quick C Ver.1.1、Microsoft C Ver.5.1、4.0、TURBO C Ver.2.0、Lattice C Ver.4.0に対応しています。(各別製品)

■グラフィックス関数は全てアセンブラで記述されており、しかもLIOをコールせずに、GDCやグラフィックスVRAMに直接アクセスしているため、究極のスピードを実現しています。6種類の描画モード、128倍迄の画面ハードコピーなどUCGならではの機能が豊富にあります。高速CADなどのアプリケーション開発用に最適です。

■ANSIの規格に対応したプロトタイプ宣言がしてあるため、返り値と引数の数と型のチェックが厳密に行えます。

※当社に直接お申し込みの方に限り、テキスト画面の拡大ハードコピーが行えるユーティリティのソース・プログラムを差し上げます。

※販売店様用に『マドレールSF』、『Ultra C Graph』のデモプログラムを用意しております。

●価格／¥32,000　●開発／株式会社　アークブレイン

開発スタッフ募集

※新卒者同時募集※見習も可

Archbrain

株式会社 アークブレイン 〒151 東京都渋谷区幡ヶ谷3-20-2 グランドメゾン幡ヶ谷207 TEL.(03)375-8968 FAX.(03)375-8767

■お申し込み・お支払い方法■

『マドレールSF』、『UCG』はパソコン・ショップまたは当社で直接お買い求めいただけます。住所・氏名・電話番号・商品名(UCGではC言語の種類も明記)・個数・ディスクの種類を明記のうえ電話、FAXまたは封書にてお申し込み下さい。お支払いは銀行振込、現金書留または郵便振替にてお願い致します。入金確認後、即日発送いたします。送料は当社で負担いたします。

《振込銀行》

八千代信用金庫	笹塚支店	当座	005-1043492
第一勧業銀行	笹塚支店	当座	161-0115523
三菱銀行	笹塚支店	普通	138-4857406

《振替口座》　東京1-251652

●MS-DOS、Microsoft C、Quick CはMicrosoft社の商標です。●TURBO CはBorland社の商標です。●Lattice CはLattice社の商標です。

図形処理新春セミナー

図形処理情報センターでは２月に東京において図形処理新春セミナーを開催いたします。図形処理新春セミナーはこれからのCAD／CAM／CAE，コンピュータ・グラフィックスの展望を切り開くタイムリーで話題豊かなキーワード・テーマを中心に開催されます。現状のテクノロジーでは問題解決が難しいと考えている先進的エンジニア，技術管理者の参加を求めます。

コンピュータ・グラフィックス

2月14日(火)15(水)　2日間　　10:00～17:00

主なテーマ(予定)

- ハイパーメディア
- 物理法則アニメーション
- レディオシティ
- メディア・テクノロジーとのインターフェース
- レイ・トレーシング
- AIとCG
- 人工知能とコンピュータ・グラフィックス
- コンピュータ・グラフィックスとホログラフィー
- フラクタルとCG
- 立体視とCG

CGの産業応用

2月17(金)　1日　　10:00～17:00

主なテーマ（予定）

- CGビデオアニメーションの作り方
- コンピュータ・グラフィックスの産業応用をどう進めるか
- プラント設計・施工におけるコンピュータ・グラフィックス
- 容器設計におけるコンピュータ・グラフィックス
- 3次元アパレルCADの最新動向

図形処理新春セミナー

CAD／CAM／CAE

2月23(木)　24(金)　2日間　　10：00～17：00

主なテーマ（予定）

- CAD／CAM／CAEのデータ交換
- CIM
- 4次元CAE
- インテリジェントCAD
- ソリッド・モデリング
- 3次元CAD／CAM／CAE
- テクニカル・イラスト
- 図面読取りと図面認識
- 動作解析
- 非線形問題
- CAD／CAMのための ワークステーション

最新のコンピュータ

2月20(月)　1日　　10：00～17：00

主なテーマ（予定）

- Mach OSの最新動向
- ニューロ・コンピュータ
- 人工知能パソコン
- ワークステーションからスーパーコンピュータまで —アーキテクチャと性能評価
- ネットワークとワークステーション

会場　東京厚生年金会館　　東京・新宿

参加料

	1名ご参加の場合	2名以上ご参加の場合(注)
2日間セミナー	38000円	34000円
1日セミナー	19000円	17000円

- 2名以上ご参加の場合とは，同一コースに同一申し込み書にて2名以上ご参加の場合です。
- 各コースとも定員になり次第，募集を締め切ります。

※詳細なパンフレットを下記までご請求下さい。

図形処理情報センター　セミナー係
〒101 東京都千代田区神田神保町1—64 神保町協和ビル　☎03(293)6161　FAX03(293)6164

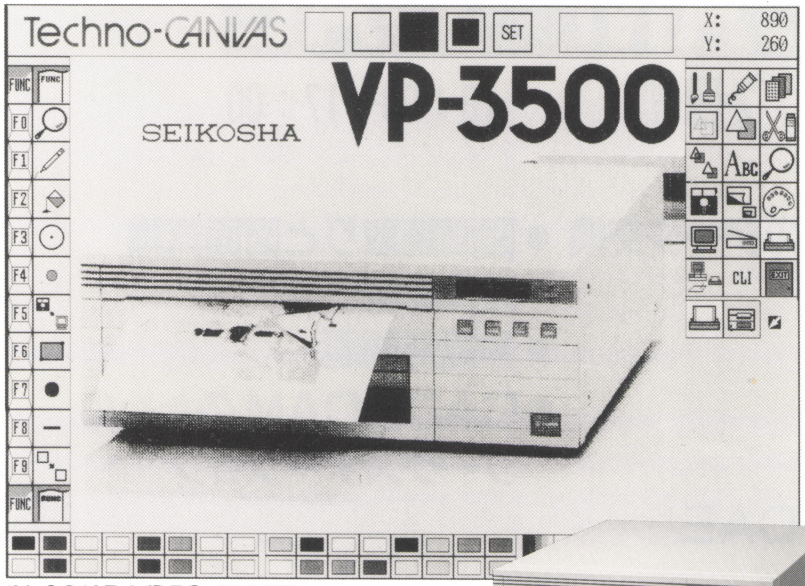
（資料請求番号　33）

広がるパフォーマンス

超・高解像度ハード・コピー。

SEIKOSHA

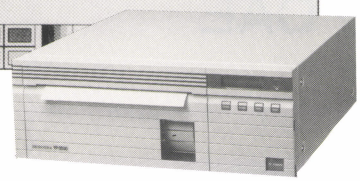
64階調で微妙なトーンまで
スピード・スチル。
より使いやすく、多目的に、
最大プリント・ラスタ数1,240本
ビデオ・プリンタ、登場。



- 最大プリント・ラスタ数1,240本の高解像度で、最大87MHzまでのビデオ信号にオート・スキャンで対応し、各社ハード・ウェアに接続可能。
- 各種ビデオ映像を鮮やかに、スピーディにハード・コピー。
- 見やすい液晶パネルを装備、操作設定はテン・キーでワンタッチ。
- 水平解像度／プリント・スピード／階調数の選択が可能。
- 1,280×1,280×6bitsのビデオ・フレーム・メモリを内蔵。

※VP-3500に関するお問合せは、下記の住所までお願いいたします。

GRAY SCALE VIDEO PRINTER
VP-3500
新製品



株式会社 精工舎

電子機器事業本部　第二販売部
〒130東京都墨田区太平4-1-1
☎(03)621-3501(代)

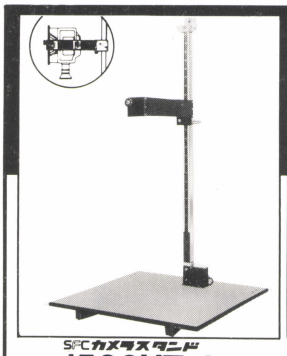
資料請求券
PIXEL 2月号
VP-3500

（資料請求番号　34）

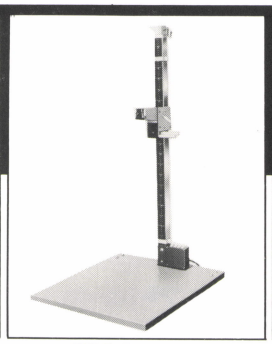
画像処理用カメラスタンドは

エス・エフ・シー・

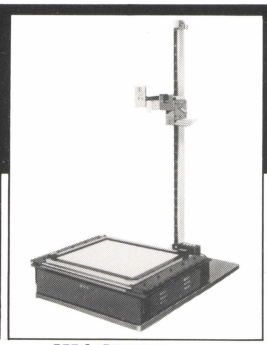
小型のCCDカメラから大型の3管式カメラまで、どんなタイプのカメラにも対応



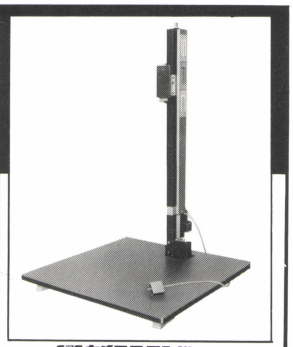
SFCカメラスタンド
1500VT-3型
カメラ正像取付型



SFCカメラスタンド
1100-V型

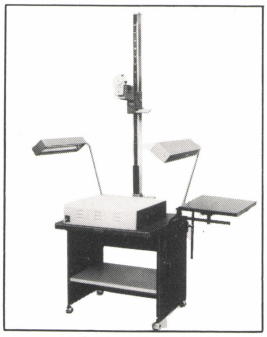


SFCカメラスタンド
1300-XY型
透過光用XYテーブル付

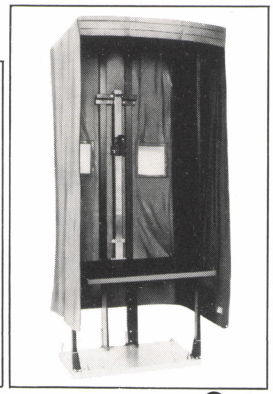


SFCカメラスタンド
1300-PD型
モータードライブ型

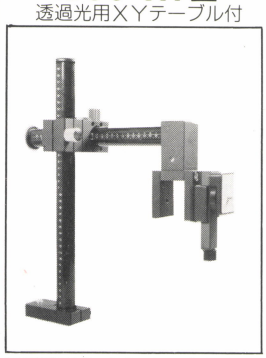
SFCビデオスタンド
フルセット型



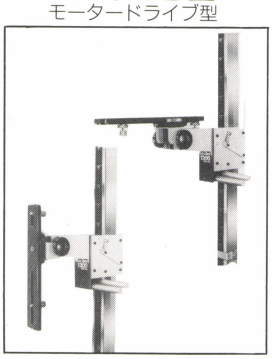
SFCオペレータースタンド
1300型
モニターラック，
CCU格納棚付



SFCオペレータースタンド **C型**
しゃ光幕付カメラスタンド
室内灯の下や窓際でのオペレートに



SFCマルチカメラホルダー
小型カメラ用



SFC VHアダプター
カメラ垂直，水平可変装置

くわしくは下記カタログをご請求下さい。

株式会社 エス・エフ・シー・　PX係　〒101 東京都千代田区神田神保町1-38 秋元ビル　TEL03(291)3781(代)　FAX03(291)3783

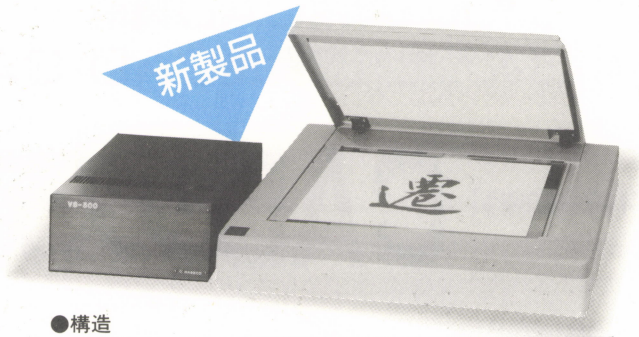
（資料請求番号　35）

挨愛始

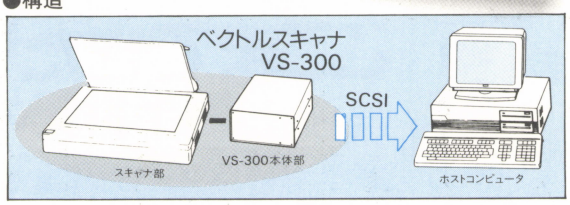
イメージ/ベクトルデータ自動変換

ベクトルスキャナ VS-300

ベクトルスキャナVS-300は、A3サイズのイメージスキャナによって読み込んだ文字や図形のイメージ情報をもとに自動的にベクトルデータ(輪郭線補間データ)を生成し、SCSIインターフェイスを介してホストコンピュータへ受渡します。



●構造



仕様 〈イメージスキャナ部〉●原稿サイズ／最大A3●読取解像度／400・200ドット/インチ●階調性／2値●大きさ／640(W)×547(D)×110(H)mm●重量／約25kg●使用電源・消費電力／AC100V50/60Hz・180W〈VS-300本体部〉●CPU／M68000(12MHz)●DRAM／4MB●イメージ・ノイズフィルター●スキャナ・インターフェイス／16ビットパラレル●ホスト・インターフェイス／SCSI●大きさ／287(W)×270(D)×147(H)mm●重量／約6kg●使用電源・消費電力／AC100V50/60Hz・100W〈データ形式〉●直線近似ベクトル●円弧＋直線近似ベクトル(オプション)●ランレングス(オプション)●MS-DOSファイルコンバータ(オプション)

※予告なしに仕様変更する場合があります。

RASSCO
ラスコ株式会社

●イメージ/ベクトルデータ自動変換ベクトルスキャナVS-300はOEM対応も可能です。

●本社/170東京都豊島区南大塚2-45-4 三栄ビル ●分室/170東京都豊島区南大塚2-39-7 ヤマモト大塚ビル TEL 03-946-8664(営業部) FAX 03-946-4389

(資料請求番号 36)

PIXEL

1989 2
No.77
1989年2月1日発行　毎月1回1日発行

75 [特集1] **新しいCG表現を可能にするメタボールとその使い方**

76 メタボールと私●高沖英二
81 メタボール実験プログラム●伊藤博文

86 [特集2] **3次元機械設計用CAD/CAM——最新技術動向とユーザーにおける使い方**

86 3次元設計用CAD概論●田中秀樹・山本理・木下篤
94 3次元機械設計用CAD/CAM/CAEユーザー調査

106 [特集3] **CADシステムのカスタム化**

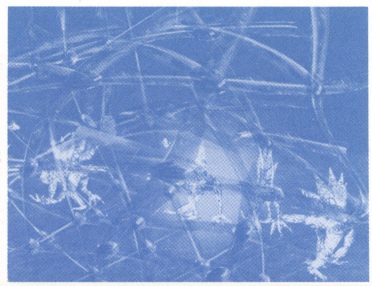
133 [特別寄稿] ここまできたリアリズム——四半世紀の進展●中前栄八郎

[トレンド]

63 今こそベンチャー企業がカスタムICを使ってGWSを作る時代だ

PIXEL別冊No.7「CAD/CAM,CG年鑑89」 昨年の別冊No.6「CAD/CAM, CG総覧88」に続いての期待の別冊ただいま発売中／ 最新の技術・製品動向，ベンダー，CGプロダクション，イベントなどCAD/CAM/CAE，コンピュータ・グラフィックス，画像処理とその応用分野における各種情報を網羅したオールマイティーなバイヤーズ・ガイドです。「総覧88」よりも情報源としての使い易さと情報量をさらにアップし，これを通してこの分野に関するあらゆる情報にアクセス可能になります。ユーザー，ベンダー，研究者を問わずお使いいただけます。ご予約はお早目にどうぞ。詳細はお知らせページ（P39）をご参照ください。

●表紙の解説



"Bubblisque"

モデリングには自作のツール，ダイナメーション・システムを使用し，多数のオブジェクトの集合体をさらに滑らかに変形させ，現実にあり得ないシャボン玉の構成物を表現してみた。レンダリングにはウェイブフロントを使用している。

最近まで，コンピュータ・グラフィックスによる作品はどことなく安っぽい感じがするのが否めなかった。しかし，最近のモデリングや質感の表現の多様性，解像度や計算可能なオブジェクト数の増大といったことにより，写真や絵画に見劣りしない（あるいはそれ以上の）画像が作成可能となってきた。

私は少年の頃，ドイツの画家マックス・エルンストが大好きでよく画集を眺め，いつかこんな絵を描きたいと夢見ていた。この作品を見ていると，コンピュータ・グラフィックスで油絵に並べられる作品がようやくできるようになるところまできたと感じられて感慨深いものがある。

㈱ハイテックラボ・ジャパン／出渕亮一朗

[注目の新製品]

176 **3次元レンダリングを高速処理するディスプレイ COMTEC3555レンダリング・ターボ** ●増田孝

[寄稿]

55 **ボリューム・レンダリング** ●柴本猛

59 **物理法則ベースモデリング** ●寺沢幹雄・柴本猛

143 **テクニカル・イラストの変遷とCAD化への可能性** ●西岡和紀

150 連載◉**CGのための図学(10)** ●長島忍

154 連載◉**Cプログラミングを用いたレンダリング・ソフトの実践シリーズ(4)** ●出渕亮一朗

171 連載◉**ラスター・グラフィックスのソフトウエア(27) レイ・トレーシング(5)** ●大野義夫

169 連載◉**GIS総論(3) 米国におけるGISの現状** ●太田和亨

47 COLOR IMAGES　66 NEW PRODUCTS　179 NEWS SCAN　105・175 BIBLIOGRAPHY

75 From The Editor's Desk　39 お知らせページ　180 掲示板　182 編集後記

広告索引

ア
アークブレイン　31
アステック　16
アストロデザイン　27，28
インテグラ　8
インフォメーション アンド コントロール研究所　24
エイ・アイ・テクノロジー　18
エス・エフ・シー　34
カ
カナレ電気　38
キャダムシステム　表3
クボタコンピュータ　17
グラフテック　2，3
小松製作所　29
サ
神鋼電機　表4
図形処理情報センター　32，33
精工舎　34
セイコー電子工業　1
ソニー　26
ソニー・テクトロニクス　表2
タ
立石電機　10
テクネ　13
東京エレクトロン　15
東芝　21
東陽テクニカ　9
東洋電機製造　12
ナ
日本・データゼネラル　11
日本電気ホームエレクトロニクス　20
日本バイナリー　23
日本無線　19
ハ
日立製作所　6，7
ヒューコム　62
ファモティク　30
マ
三井造船システム技研　14
武藤工業　4，5
ラ
ラスコ　35
リコー　25
ワ
ワイ・デー・ケー　22

図形処理新春セミナー　今回の開催は2月に決定いたしました。新春セミナーではこれからのCAD/CAM/CAE，CGの展望を切り開くタイムリーで話題豊富なテーマを中心に，CG，CGの産業応用，CAD/CAM/CAE，最新のコンピュータの4つのコース別で行います。詳細は広告(33，34ページ)をご参照ください。

ビデオで見るCG最前線II(日本語版)　今春販売開始し，大好評いただいた「ビデオで見るCG最前線」(ACM/SIGGRAPH監修)の日本語版が出来ました。内容も英語版未収録のものも加わり，さらに充実。CGの最新情報は「……II」日本語版で！　詳しくはお知らせページ(40ページ)をご覧ください。

CANARE

配線スッキリ。

同軸ケーブルがマルチになった。
カナレ同軸マルチケーブル。

特性インピーダンス：75Ω

画像情報機器の進歩に伴って、従来1本の同軸ケーブルで伝送していたものが、最近では3～5本で伝送するようになりました。その結果、機器周辺はケーブルがいっぱい…。
そこでカナレでは同軸マルチケーブルを開発。機器周辺の配線をスッキリとさせます。
各回線はカラー同軸ユニットですから、RGB信号等を容易に識別でき、確実で手早い接続が可能です。さらに、リターン・ロスは1GHzまで25dB以上と高性能、減衰量は3C-2VSと同等、CG機器、デジタル放送機器、ビデオプロジェクター等にご利用いただけます。

※コネクタ付ケーブルの製作も承ります。下記までお問い合わせください。

仕様

V3-3C (3C-2VS相当3本)　V4-3C (3C-2VS相当4本)　V5-3C (3C-2VS相当5本)

トータル高周波エンジニアリングをめざす。

設計 → 製造 → ハーネス加工

"コネクタに最適なケーブルを。
ケーブルに最適なコネクタを。"
設計から製造、ハーネス加工まで
カナレならお応えできます。

カナレ電気株式会社

●東京本社/☎03(635)8241㈹ 〒130 東京都墨田区両国4丁目37-2 TKF会館4F
●大阪営業所/☎06(229)8021㈹ 〒541 大阪市東区伏見町2丁目19-2 Jビル4F
●福岡出張所/☎092(524)6951 〒810 福岡市中央区平尾5丁目3-17
●名古屋本社/☎05616(2)5446㈹ 〒480-11 愛知県愛知郡長久手町大字長湫字段之上1-2

PIXEL別冊号NO.7　昨年に続く，第二弾！

CAD/CAM/CAE，コンピュータ・グラフィックス（CG），画像処理の最新記事，製品動向，メーカー一覧，役に立つ各種の資料をこの1冊に集成しました。これ1冊で技術，応用，製品の最新動向がわかります。

NEW

購読申込み受付け中！

89年度版　好評発売中

CAD/CAM/CG年鑑89

［本書の特長］

- CAD/CAM，CAE，コンピュータ・グラフィックス，画像処理の分野全体にわたり，最新の動向が把握できるように編集されています。
- 営業から技術，企画まで企業のいろいろなセクションの人に，また研究や教育関係の人にも必要な内容になっています。
- 製品関係の記事およびメーカー・ベンダーの一覧により，製品バイヤーズガイドとしても役に立ちます。
- 資料編集は電話帳代わりにもなりますので，座右の書として便利です。

記事編

CAD/CAM/CAE，画像処理とその分野で使用されるシステム，機器，ソフトウエアなど最新の製品動向の解説とそれらの製品名を掲載します。

- 展博映像におけるコンピュータ・グラフィックス
- 専門学校におけるCG教育カリキュラム
- 安全性検証のためのシミュレーション
- 印刷における画像処理
- 動画像処理の最新動向
- 3次元メディカルにおける画像技術応用
- 図面自動読取りの最新技術動向
- 地理情報処理の最新動向
- 最新の分子設計と薬物設計
- 最新のテクニカル・イラストレーション
- デジタル・フォントの最新動向
- 人体モデルの動向
- 機械設計におけるデータ交換
- 金型のCAEの最新動向
- 機械設計における最新のプリ・ポスト動向
- 3次元アパレルCADの最新動向
- インテリジェントCADの機械設計への応用
- 建築におけるCADとプレゼンテーション
- 自動車デザインにおけるCG
- グラフィック・デザインにおけるCG応用
- 放送におけるコンピュータ・グラフィックス
- 景観設計におけるコンピュータ・グラフィックス応用動向
- 最新のサイエンティフィック・ビジュアライゼーション
- パッケージ・デザインにおけるコンピュータ・グラフィックス

製品編

CAD/CAM/CAE，CG，画像処理についての最新技術および活用方法の解説，分野ごとの技術動向，使い方など第一線の専門家がわかり易く解説します。

- ペイント
- CG用フレーム・バッファ
- パーソナルCG
- 画像処理
- パソコン画像処理
- 印刷用画像処理
- デスクトップ・パブリッシング
- 地図情報処理
- 分子設計
- カラー・ハードコピー
- 機械用CAD CAM
- 機械用CAE
- 建築用CAD
- PCB用CAD CAM
- LSI用CAD CAM
- シリコン・コンパイラ
- アパレル・テキスタイル用CAD
- パーソナルCAD
- レンダリング・アニメーション
- ワークステーション
- 32ビット・パーソナルコンピュータ
- グラフィック・アクセラレータ
- グラフィック・ディスプレイ
- デジタイザ　ダブレット
- 図面自動読取りシステム
- スキャナ
- ペン・プロッタ
- 静電プロッタ

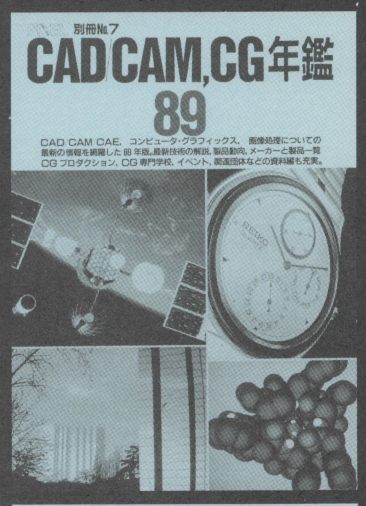
資料編

CAD/CAM/CAE，CG，画像処理の分野について，問合せをしたい，どこに依頼すれば良いのか——の時などすぐに役に立つ資料ばかりを集めました。

- CGプロダクション一覧
- 学校一覧
- イベント・CG展などの一覧
- 関連団体一覧

メーカー・ベンダー編

メーカーやベンダーの社名，住所，扱い製品名とその内容などを掲載します。製品購入や問合せの際に役に立ちます。



サイズ●A4判
ページ●204ページ
定価●2,800円

※購読のお申込みは巻末のはがきをご利用ください。

編集・発行●図形処理情報センター　☎03(293)6161　Fax03(293)6164
〒101 東京都千代田区神田神保町1-64神保町協和ビル

ビデオで見るCG最前線II
日本語版

図形処理情報センターではACM/SIGGRAPH監修によるコンピュータ・グラフィックスの解説ビデオカセット「ビデオで見るCG最前線II」(日本語版)を販売いたします。
このビデオカセットは，今春発表し大好評いただいた「ビデオで見るCG最前線」の内容に未収録の，最新機器・最新技術を60分にまとめたものです。
「ビデオで見るCG最前線」を既にご購入の方は「ビデオで見るCG最前線II」をお求めいただくことによって今夏開催されたSIGGRAPH'88で新たに登場した製品や論文発表が行われた新技術，最新のCGアニメーションがご覧いただけます。
ぜひ「ビデオで見るCG最前線」と合わせてお買い求め下さい。
「…II」においては日本語によるナレーションと日本語テキストが加えられていますのでCGの最新情報をより少ない労力で入手することができます。

収録内容

❶高性能プロセッサ
- AT&T PIXEL MACHINE 964d
- STELLAR GS1000
- ARDENT TITAN
- APOLLO DN10000
- SILICON GRAPHICS IRIS 4D/GTX
- ALLIANT VFX/80
- HEWLETT-PACKARD 825TurboSRX

ほか最新鋭機器の解説をリアルタイム操作を交えながら行います

❷レンダリング
- HP TurboSRX上のラジオシティ法
- PIXAR RenderManのアクセラレータ(プロトタイプ)
- Sun TAAC-1によるボリューム・レンダリング

❸カラー・デスクトップ・パブリッシング
- CROSFIELD STUDIO 875
- PANSOPHIC STUDIOWORKS
- LINOTRONIC L300
- AVALON PHOTOMAC

❹カラー入出力周辺機器
- 入力スキャナ
- 熱転写プリンタ
- 融溶型熱転写プリンタ
- インクジェット型プリンタ
- カラー複写機

各方式の最新機器解説

❺ビデオ
- FOLSOM RESEARCH AURORA/300
- PARALLAX 1280
- D2対1インチNTSC

❻サイエンティフィック・ビジュアリゼーション
- カール・マッコーバー
- チャールズ・スーリ
- ドナ・コックス
- ジム・ブリン
- ベノワ・マンデルブロ

ら豪華な解説陣により科学技術結果の視覚化の最新動向をまとめます

監修
◉ACM/SIGGRAPH

価格
◉VHS(家庭用1/2インチ) **45000**円
※VHS(60分　全1巻)　※日本語解説書および製品の仕様(英文)が付属します　※ナレーションは日本語です

制作
◉パシフィック・インタフェース　デュポン

お問合せ先●図形処理情報センター　SIGGRAPHビデオ係
〒101 東京都千代田区神田神保町1-64 神保町協和ビル　☎03(293)6161　FAX03(293)6164

本ビデオカセットをお申し込みの方は，下の申込用紙にご記入の上，上記の住所へ郵送もしくはファクシミリにてご送付下さい。受付後，請求書をお送りいたします。

「ビデオで見るCG最前線II」申込書

※コピーでも構いません。　　年　月　日

氏名(フリガナ)	
勤務先・学校名	
所属部門名	
住　所	〒　　電話番号　(　)　—
巻数／価格	巻× **45000** 円=　　円
備　考	

8812

図形処理情報センターの海外技術視察団

NCGA'89とCAD/CAM, コンピュータ・グラフィックス 米国最新技術視察団

募集中

視察団への参加募集

図形処理情報センターではCAD/CAMおよびコンピュータ・グラフィックスの産業応用に関する権威ある大規模なイベントであるNCGA'89が4月に開催されるのに合わせて，このイベントを視察すると同時に最先端の技術に直に触れる研究機関・企業訪問を行う視察団の派遣を企画しております。

CAD/CAM,コンピュータ・グラフィックスの分野における代表的な研究者にコーディネーターとして同行・指導いただき，各訪問先には通訳が同行いたしますので，初心者の方でも安心して参加できます。

長年この分野で実績をもつ当センターならではの企画です。ぜひご利用下さい。

本視察団の特長

○NCGA, SIGGRAPHなどCAD/CAM, コンピュータ・グラフィックスの国際会議/展示会への技術視察団派遣に長年の実績をもっておりますので，安心してご参加いただけます。

○一流の研究者がコーディネーターとして同行・指導，さらに通訳が同行いたしますので，初心者の方でも安心して参加できます。

○コンピュータ・グラフィックス，CAD/CAMの専門誌であるPIXEL編集部が選んだ，今最も注目されている大学・研究機関，関連企業を視察できます。個人ではなかなか行きにくい先端企業・研究機関ですので，日本で得られない最新の情報が短期間で効率的に収集できます。

○NCGAに長年参加してきたノウハウ，経験から，展示会の特長，見どころなどを十分に把握しているため，参加者のご要望などに十分にお応えできます。

NCGA'89とは

米国のコンピュータ・グラフィックス，CAD/CAM/CAEに関する団体であるNCGA(National Computer Graphics Association)が主催する，コンピュータ・グラフィックスとその応用についての大規模な国際会議および機器展示会です。NCGA'89は4月17日から20日まで，米国ペンシルベニア州フィラデルフィアで開催され，最新の論文発表，実用的なセミナー，200社以上の大規模な製品展示，コンピュータ・グラフィックス映像の発表が行われます。また，CGM, PHIGS, CALS, MAP/TOPといったグラフィックスやCIMの標準に関する展示会であるINTEGRATE'89も隣接会場で開催されますので，グラフィックス標準化に関心をおもちの方は非常に効率的に情報収集ができます。

旅行期間　1989年4月10日(月)～23日(日)

視察訪問先　(以下の候補から4～5ヶ所を訪問する予定です)

○Wavefront Technologies
ウェーブフロント・テクノロジーズ
○California Institute of Technology
カリフォルニア工科大学(カルテック)
○Symbolics Graphics Division
シンボリックス グラフィック事業部
○University of Utah
ユタ大学
○Brigham Young University
ブリガム・ヤング大学(BYU)
○NCSA (National Center of Supercomputing Applications)
米国スーパーコンピューティング・アプリケーション研究所
○University of Illinois
イリノイ大学
○Sun Microsystems
サン・マイクロシステムズ
○NASA Ames Research Center
NASAエイムズ研究所
○AutoDesk
オートデスク
○Apple Computer
アップルコンピュータ
○University of North Carolina
ノースキャロライナ大学
○Numerical Designs
ニューメリカル・デザインズ
○Massachusetts Institute of Technology
マサチューセッツ工科大学
○Prime Computer
プライム コンピュータ

企画◉図形処理情報センター　主催◉日本交通公社　団体旅行新宿支店

□企画内容についてのお問合せ先
図形処理情報センター　担当：松尾
〒101 東京都千代田区神田神保町1-64 神保町協和ビル
☎03(293)6161　IFAX03(293)6164

□お申し込み・お問合せ先
日本交通公社　団体旅行新宿支店〈運輸大臣登録一般旅行業第64号〉
担当：手塚　(営業第6課井上グループ)
〒160 東京都新宿区西新宿1-18-8 新宿スカイビル4F　☎03(346)0181

CAD/CAMアルファ シリーズ

CAD/CAMアルファ・シリーズは，CAD/CAM/CAEの基礎技術，最新の話題，運用のノウハウなどについて，いま最も必要としている記事を満載しています。

各号は大きなテーマごとに編集され，初心者から経験者まで，ユーザーからシステム開発者まで，どの層の方にもお読みいただけます。

No. 1 大特集 どうなる，これからのCAD/CAMとCAE!

最近のパーソナルCADとエンジニアリング・ワークステーション
エンジニアリング・ワークステーションとソフトウエア
パーソナルCADと分散処理
画像処理の製造業への応用

一般
- やさしいCAD/CAM入門の入門
- わが国におけるCAD/CAM導入の現状
- CAD/CAMシステムの研究

No. 2 大特集 いま最も注目されているパソコンCADの実力と使い方

ユーザー事例
- ミサワCADシステム（建築）
- 岡村製作所（機械設計）
- リコー（LSI設計）
- 大信工業，重野製作所（金型）
- プラスチック金型設計

新領域に臨むパソコンCAD
- ハンドヘルドCAD
- パソコンに接続する小型モデリング・マシン

シリーズ・アプリケーション
- キャノン　東芝　日立　ブリヂストン

A4変型判／定価各号1,500円　送料当社負担

PIXEL別冊

アプリケーション

コンピュータ・グラフィックス ユーザー事例集

製造業におけるCAD／CAMを中心に，アニメーション，建築・土木，構造解析，マッピング，リモートセンシング，アパレル などの各分野における代表的ユーザーを1冊に収録しました。これから導入を検討されている企業，機能アップを考えている企業にとって見逃がせない1冊となります。

- 東洋工業におけるCAD/CAM
- 自動車部品へのCAD/CAMの適用
- 航空機内装品生産におけるCAD/CAMシステムの利用
- 石川島播磨重工業での船殻CAD/CAMの現状
- 日本建鉄の金型加工におけるCAD/CAMシステムの適用事例
- 立石電機におけるCAD-CADAMシステムの紹介
- CAD一貫システムへの取組みとその事例
- 冷間コイルばね成型へのCAE導入
- 光学設計分野におけるCAD/CAM
- 有限要素法解析における図形処理の一例
- コンピュータ・グラフィックスの分散統合処理へのアプローチ
- プラント機械設計用CADシステムの開発と適用
- 電設メーカーにおけるCAD/CAM適用例
- 集積回路設計におけるコンピュータ・グラフィックスの利用
- 電子回路基板におけるCAD/CAMの適用
- 日立電子におけるCAD/CAMの適用
- プリント配線基板の設計・製造へのCAD/CAMの導入事例 他

定価　4,500円

グラフィック・システムを効率的に使いこなすためには，他社での事例を研究することがきわめて重要です。本書は導入の検討，システムの設計と選択，標準化などの受入れの準備，実際の運用，評価と今後の展望――などについて，さまざまな業種・分野からの事例報告をまとめました。

お申込みは巻末の葉書をご利用下さい。

※お申込みは巻末のはがきをご利用ください。

編集◉発行／図形処理情報センター
〒101 東京都千代田区神田神保町1-64神保町協和ビル
☎03(293)6161　Fax03(293)6164

PIXEL 別冊 No.6

CAD/CAM,CG総覧88

CAD/CAM,コンピュータ・グラフィックス,画像処理の最新技術,システム・機器の製品一覧,役に立つ資料など満載

記事編

CAD/CAM/CAE, CG, 画像処理の基礎技術と応用分野について,技術動向,使い方の動向をやさしくまとめました。

●CAD/CAM,CG用汎用コンピュータ・画像生成用コンピュータ・システム●ハードコピー機器●建築CAD●工業デザイン用CAD●レンダリングとアニメ●地図情報処理●など

製品編

CAD/CAM/CAE, CG, 画像処理とその応用分野で使用されるシステム,機器,ソフトウエアなどを収録,解説。

●CAD/CAM, CG用コンピュータ●グラフィック・ディスプレイ装置●デジタイザ/スキャナ●各種CAD/CAMシステム●CG用システム,ハードウエア,ソフトウエア●など

資料編

CAD/CAM/CAE, CG, 画像処理の分野で仕事に役立つ資料を集めました。

●CGプロダクション一覧●CGクリエイター,研究者一覧●学校一覧●イベント,CG展一覧●関連団体一覧

メーカー・ベンダー編

メーカーやベンダーの社名,住所,扱い製品を収録。

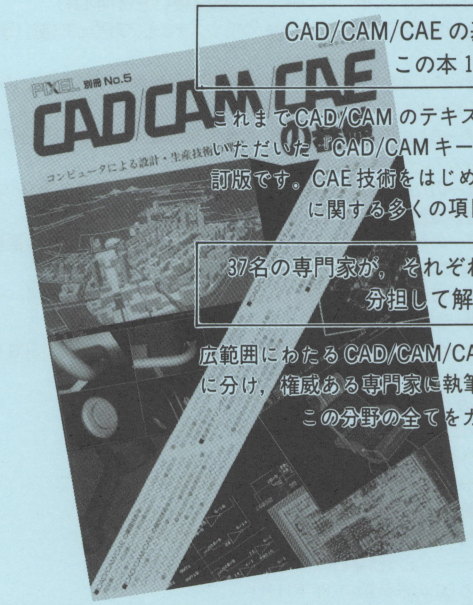
A4判／204ページ／3,500円

コンピュータによる設計・生産技術の理解

PIXEL 別冊 No.5

CAD/CAM/CAEの基礎

CAD/CAM/CAEの入門書・教科書として最適!



CAD/CAM/CAEの基礎的な知識はこの本1冊で十分です。

これまでCAD/CAMのテキストとしてご好評をいただいた『CAD/CAMキーワード』の全面改訂版です。CAE技術をはじめ,設計・生産技術に関する多くの項目を追加しました。

37名の専門家が,それぞれの専門分野を分担して解説しています。

広範囲にわたるCAD/CAM/CAE分野を36項目に分け,権威ある専門家に執筆を依頼しました。この分野の全てをカバーしています。

第1章 CAD/CAM/CAEの基礎技術
CAD/CAM/CAE概論　CAD概論　CAM概論など

第2章 CAD/CAM/CAEの個別技術
統合化された設計と生産　CIM　FA, FMS
ロボット　数値制御（NC）など

第3章 形状モデリングとコンピュータ・グラフィックス
形状モデリング　図形表示のための変換
コンピュータ・グラフィックス　レンダリング技術など

第4章 CAD/CAM/CAE システム
CAD/CAMシステムのさまざまな形態
コンピュータ・システム
パーソナルコンピュータによる
CAD/CAMなど

定価○2,800円(送料無料)

A4変形判○264ページ

※お申込みは巻末のはがきをご利用ください。

編集◉発行／図形処理情報センター
〒101 東京都千代田区神田神保町1-64神保町協和ビル
☎03(293)6161　Fax03(293)6164

PIXELバックナンバー/分野別一覧

※お申込は巻末のハガキにてお申込みください。

定価/980円
(88年1月号のみ 1,280 円)

※送料がかかる場合もあります

各タイトル後(　)内は号を表わします。
例:(87年1月)は1987年1月号

●

(特集):特集記事
(企画):企画記事
(連載):連載記事
(トレンド):トレンド
(シリーズ):シリーズ

CAD/CAM 関連記事

(連載)講座:CAE のための数値解析入門(87年1月～12月)
(連載)プリント基板 CAD/CAM の現状(87年1月～2月)
(連載)市場に即応する生産システムを目指して FMS の現状(87年1月～4月)
(連載)これからの CAE 入門(87年5月～11月)
(連載)金型専業メーカーにおける実践的 CAD/CAM 活用事例 I
(連載)金型専業メーカーにおける実践的 CAD/CAM 活用事例 II (87年9月)
(連載)金型専業メーカーにおける実践的 CAD/CAM 活用事例 III (87年10月)
(連載)金型専業メーカーにおける実践的 CAD/CAM 活用事例 IV (87年11月)
(連載)金型専業メーカーにおける実践的 CAD/CAM 活用事例 V (87年12月)
(シリーズ特集)CAD/CAM, CG のニュー・アプリケーション(87年7月)
(シリーズ特集)CAD/CAM, CG のニュー・アプリケーション(87年9月)
(シリーズ特集)CAD/CAM, CG のニュー・アプリケーション(87年10月)
(特集)コンピュータ映像のいろいろ(88年1月)
(シリーズ企画)ユーザーのための CAD/CAM/CAE システム入門
　第1回　機械設計用 CAD/CAM 入門(前編)(88年1月)
　　機械設計用 CAD/CAM 入門(後編)(88年2月)
　第2回　電子・電気回路設計用 CAD/CAM/CAE システム(88年2月)
　第3回　機械設計用 CAE と構造解析(88年3月)
　第4回　建築用 CAD システム(88年4月)
　最終回　金型設計用 CAD/CAM/CAE (88年5月)
(特集)CAD/CAM, CG 画像処理を支える日本の研究最前線(前)(88年 4 月)
(特集)CAD/CAM, CG 画像処理を支える日本の研究最前線(後)(88年5月)
(特集)インテリジュント CAD は現在の CAD をどう変えるか?(88年6月)
(特集)機械設計用 CAE と構造解析(88年8月)
(企画)計算シミュレーションの世界　ここまできた未来予測技術(前)(88年10月)
(企画)計算シミュレーションの世界　ここまできた未来予測技術(後)(88年11月)
(企画)IGES って何だろう(88年9月)
(トレンド)異なる CAD/CAM 間のデータ交換は不満だが IGES を使う(88年6月)
(トレンド)ミシガン大学における CAE ネットワーク(88年7月)
(トレンド)パソコン CAD のデータ交換標準案 PCES (88年7月)
(トレンド)CAD/CAM のデータ交換に対するメーカーの取り組み(88年8月)
(トレンド)パーソナル CAD データ互換のための中間ファイル PCES が10月を目途に製品化(88年9月)
(トレンド)CAD システムの新しい将来像「属性モデリング」とはなにか(88年10月)
(トレンド)製品モデルの交換仕様 STEP の動向　いつ IGES が不用になるのか?(88年11月)
(トレンド)最新のインテリジェント CAD の最新動向(88年11月)

ハードウエア,ソフトウエア,システム製品

(企画)パーソナル・ソリッド・モデリング「マドレール」の機能と使い方(87年1月)
(特集)ワークステーションで利用できる CAD/CAM/CAE システムとソフトウェア(前)(後)(87年2月3月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理用機器・システム①(エンジニアリング・ワークステーション)(87年5月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理用機器・システム②(グラフィック・ディスプレイ)(87年6月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理用機器・システム③(デジタイザと図面読取装置)(87年7月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理 用機器・システム④(画像処理用機器・システム)(87年8月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理用機器・システム⑤(ハードコピー機器・装置)(87年9月)
(シリーズ企画)ユーザーのための CAD/CAM, CG, 画像処理用機器・システム⑥(CG 用ハードコピーとソフトウェア)(87年10月)
(特集)今、どんな製品が売れているか(87年12月)
(トレンド)ステラ・コンピュータ社長/アーサー・カー氏インタビュー(88年6月)
(トレンド)NCGA '88 ハイライト製品(88年6月)
(トレンド)ワークステーションが CAD/CAM, CG を変える(88年6月)
(トレンド)米国の最新 CG 製品紹介(88年8月)
(トレンド)分子グラフィックス学会年次大会の概要決まる(88年8月)
(特集)最新のワークステーション(88年8月)
(特集)SIGGRAPH '88 と CG の最新動向(88年11月)

PIXELバックナンバー/分野別一覧

コンピュータ・グラフィックス

(特集)コンピュータ・グラフィックス小史(前編)(87年1月)
(特集)コンピュータ・グラフィックス小史(中編)(87年3月)
(特集)コンピュータ・グラフィックス小史(後編)(87年8月)
(連載)フランスのコンピュータ・グラフィックス(87年1月～4月)
(連載)コンピュータ・グラフィックスによる光のシミュレーション(87年1月)
(連載)コンピュータ・グラフィックス入門(87年1月～12月)
(特集)SIGGRAPH '86-米国コンピュータ・グラフィックスの最新動向(II)
最も注目されたCGシステム・5選(87年1月)
(連載)メディア・テクノロジ(87年1月～3月)
(特集)パーソナルコンピュータによる新しいアイデアのCG(87年2月)
(企画)プロのCG映像ができるまで(87年2月)
(連載)コンピュータ・グラフィックス(87年2月～11月)
(連載)ラスター・グラフィックスのソフトウェア(87年3月～1月)
(企画)CG年賀状(87年3月)
(シリーズ特集)CG映像プロダクションとその作品①(87年4月)
(シリーズ特集)CG映像プロダクションとその作品②(87年5月)
(シリーズ特集)CG映像プロダクションとその作品③(87年6月)
(シリーズ特集)CG映像プロダクションとその作品④(87年7月)
(シリーズ特集)CG映像プロダクションとその作品⑤(87年8月)
(シリーズ特集)CG映像プロダクションとその作品⑥(87年9月)
(シリーズ特集)CG映像プロダクションとその作品⑦(87年10月)
(特集)SIGGRAPH '87報告(87年10月)
(特集)最新のグラフィック・エンジンとレンダリング・システム(87年11月)
(シリーズ企画)ユーザーのためのCAD/CAM, CG, 画像処理用機器・システム⑥
CG用ハードウエアとソフトウエア(87年10月)
(特集)コンピュータ映像のいろいろ(88年1月)
(特集)日本のCAD/CAM, CG研究(88年2月)
(企画)コンピュータ・グラフィックスの原点SKETCHPADとその果たした役割(88年3月)
(特集)最新のレンダリング技術とCG映像(88年3月)
(特集)リアルなレンダリング・アプリケーション(88年6月)
(特集)CGアニメーションのいろいろと作り方(88年7月)
(特集)誰にも出来るパーソナル・グラフィックスのためのいろいろツール(88年7月)
(特集)CGによる自然物のリアルな表現(88年8月)
(企画)コンピュータ・グラフィックスによるデザインの進め方(88年9月)
(特集)CGによるリアルな映像生成入門(88年10月)
(トレンド)コンピュータ映像制作の現状とこれから(88年6月)
(トレンド)PIXARが3次元レンダリングの標準RenderManを提案(88年8月)
(トレンド)39,600円でCGができる？　SIGの挑戦(88年9月)
(トレンド)実用化に向かいつつある最新のレイ・トレーシング・ソフトウェア(88年9月)
(トレンド)SIGGRAPPH '88　CG実用化へのスタート(88年10月)
(トレンド)レイ・トレーシングをどう使うか？　(88年10月)
(トレンド)SIGGRAPH '88　CG実用化へのスタート(88年10月)
(トレンド)大手ゲームメーカーのナムコがCG映像ビジネスに参入(88年11月)
(トレンド)スーパーコンピュータにおけるビジュアリゼーション(88年11月)
(特集)SIGGRAPH '88とCGの最新動向(88年11月)
(連載)CGのための図学(88年4月)

CAD/CAM/CAE, コンピュータ・グラフィックス, 画像処理共通

(特集)NICOGRAPH '87のハイライト論文(87年12月)
(特集)日本のCAD/CAM, CG研究(88年2月)
(特集)CAD/CAM, CG画像処理を支える日本の研究最前線(前)(88年4月)
(特集)CAD/CAM, CG画像処理を支える日本の研究最前線(後)(88年5月)
(特集)リアルなレンダリング・アプリケーションI(88年5月)

CG, 画像処理の応用

(シリーズ特集)CAD/CAM, CGのニュー・アプリケーション(87年8月)
(シリーズ特集)CAD/CAM, CGのニュー・アプリケーション(87年11月)

画像処理関連記事

(シリーズ企画)ユーザーのためCAD/CAM, CG, 画像処理用機器・システム④(87年8月)
(連載)新しい画像処理(87年10月)
(連載)新しい画像処理(87年11月)
(特集)コンピュータ映像のいろいろ(88年1月)
(特集)日本のCAD/CAM, CG研究(88年2月)
(特集)CAD/CAM, CG画像処理を支える日本の研究最前線(前)(88年4月)
(特集)CAD/CAM, CG画像処理を支える日本の研究最前線(後)(88年5月)

地図情報処理

(特集)期待される地図情報処理①――最新システムとその使い方(前)(87年3月)
(特集)期待される地図情報処理②――最新システムとその使い方(中)(87年4月)
(特集)期待される地図情報処理③――最新システムとその使い方(後)(87年5月)
(特集)期待される地図情報処理④(87年6月)
(特集)期待される地図情報処理⑤(87年7月)
(特集)期待される地図情報処理⑥(87年8月)
(特集)期待される地図情報処理⑦(87年9月)
(特集)期待される地図情報処理⑧(87年10月)
(特集)期待される地図情報処理⑨(87年11月)
(特集)期待される地図情報処理⑩(88年4月)
(特集)期待される地図情報処理⑪(88年10月)
(特集)コンピュータ映像のいろいろ(88年1月)
(特集)日本のCAD/CAM, CG研究(88年2月)
(特集)最新の地図情報処理システム(88年9月)

DTP

(特集)コンピュータ映像のいろいろ(88年1月)
(企画)電子出版のためのデジタルフォント(88年4月)
(トレンド)急転開するデスクトップ・パブリッシング(88年7月)
(トレンド)　DTPのための高品位日本語デジタルフォント(88年8月)
電子出版のためのPost Script入門(87年8月)

図面自動入力

(特集)図形の自動読み取りと認識, 理解(88年7月)
(特集)図面読み取りシステムとその機能(88年11月)

その他

(特集)CAD/CAM, CGのベンチャー企業(前)(87年1月)
(特集)CAD/CAM, CGのベンチャー企業(後)(87年3月)
(連載)イメージで理解する人工知能入門(87年1月～5月)
(連載)ハッカーと電子犯罪(87年5月)
(連載)情報積算法入門(87年10月)
(連載)フランスのコンピュータ・グラフィックス(87年1月～4月)
(連載)コンピュータ・グラフィックスによる光のシミュレーション(87年1月)
(連載)コンピュータ・グラフィックス入門(87年1月～12月)
(連載)講座：CAEのための数値解析入門(87年1月～12月)
(連載)プリント基板CAD/CAMの現状(87年1月)
(連載)イメージで理解する人工知能入門(87年1月～5月)
(連載)メディア・テクノロジ(87年1月～3月)
(連載)標準化(87年2月)
(連載)市場に即応する生産システムを目指して　FMSの現状(87年2月～4月)
(連載)コンピュータ・グラフィックス(87年2月～11月)
(連載)これからのCAE入門(87年5月～11月)
(企画)コンピュータ・グラフィックスの原点SKETCHPADと
その果たした役割(88年3月)
(連載)サンタモニカ通信(88年1月～2月)
(連載)ぴくせる・ぷれいく(88年6月・8月)
(連載)流れのシミュレーション(88年7月～)
(連載)展示用特殊映像の種類と歴史(88年10月)

第1回PIXEL CGグランプリ
作品募集

図形処理情報センターではコンピュータ・グラフィックス映像におけるデザイン，技術の水準を向上させ，その限りない映像表現力を追求するとともに，アートやプレゼンテーション，デザイン，ビジュアル・シミュレーションなどの分野に大きな刺激を与えることを目的に，第1回PIXEL CGグランプリを行います。
第1回PIXEL CGグランプリはコンピュータ・グラフィックスおよび画像処理による静止画の作品を対象として，応募作品の中からグランプリ作品および優秀作品をPIXEL誌上において掲載し，その栄誉を讃えます。プロ，アマおよびコンピュータの機種を問わず，どなたでも応募できますので，コンピュータ・グラフィックスによる作品制作を進めている多くの方の応募を求めます。
コンピュータ・グラフィックスに新風を吹き込むような意欲的な作品を期待します。

応募要項

コンピュータ・グラフィックスおよび画像処理を使用したアート，プレゼンテーション，デザイン，ビジュアル・シミュレーションなどの分野における静止画作品で，未発表のものに限ります。
作品はデザイン，技術，アイデア，新規性などの点から審査し，グランプリ作品および優秀作品を選定し，PIXEL誌上で掲載，表彰します。

締切り●1989年1月20日（PIXEL CGグランプリ事務局）必着

入賞作品●（入賞作品には賞状と賞金が授与されます）
グランプリ賞◘応募作品の中から特に優れた作品をグランプリ作品として表彰します。
優秀賞◘デザイン，技術，アイデアなどの点で優秀または新規性があり，作品としての完成度の高い作品を優秀作品として表彰します。

入賞作品の発表●グランプリ作品および優秀作品はPIXEL 89年4月号にて発表，表彰いたします。

応募規定●
・コンピュータ・グラフィックスおよび画像処理を使用した作品で，アート，プレゼンテーション，デザイン，ビジュアル・シミュレーションなどの分野において映像表現力を追求した静止画を対象とします。
・未発表の作品に限ります。
・作品の全部または一部にコンピュータを使用していることが必要です。
・一人何点でも応募できますが，一点一点独立したテーマのものに限ります。
・類似する絵柄は応募者で選択の上，一点に絞って応募下さい。
・プロ・アマおよび職業，国籍を問わず，どなたでも応募できます。
・作品制作に使用するハードウェア，ソフトウェアの機種は問いません。どのようなシステムを使用して制作した作品でもかまいません。
・作品はモノクローム（白黒など），カラーどちらでも結構です。
・作品としての完成度が必要です（絵柄が途中から切れているもの，ブラウン管の縁が写っているものなどは審査の対象からはずされることがあります）。
・個人での応募のほか，グループでの応募もできます。

応募される作品の形態●カラーポジフィルム（インスタント・カラーポジフィルムは印刷に適しませんので，応募できません）
・白黒の場合はキャビネ判～四切サイズのプリント
・グラフィック・プリンタなどによる出力原図（最大A3サイズまで）

応募方法●
・応募作品ごとに別紙のエントリー・シートをつけて応募して下さい。
・作品は必ず保護して書留，速達，宅配などで送付ください。
（送付過程での破損等については責任をもちかねますので梱包には注意してお送りください）
※応募いただいた作品は原則としてお返しいたしません。

主催●図形処理情報センター

協賛●（11月21日現在の協賛企業は以下の通りです。順不同）
●旭テクノコンピュータ㈱ ●アストロデザイン㈱ ●㈱エス・アイ・ジー ●エイ・アール・シー・ヤマギワ㈱
●クボタコンピュータ㈱ ●㈱テクネ ●NKK ●日本シンボリックス㈱ ●ソニー・テクトロニクス㈱
●㈱リンクス

審査●PIXEL CGグランプリの応募作品は次の審査員によって審査され，入賞作品が決定されます。（順不同）
●ネルソン・マックス氏／Hikari Kinema社CGコ・ディレクター
●藤幡正樹氏／㈱フロッグス主宰
●三井秀樹氏／筑波大学 芸術学系助教授
●木村卓氏／㈱リンクス 制作部 チーフデザイナー&ディレクター
●河内隆幸／図形処理情報センター PIXEL編集長

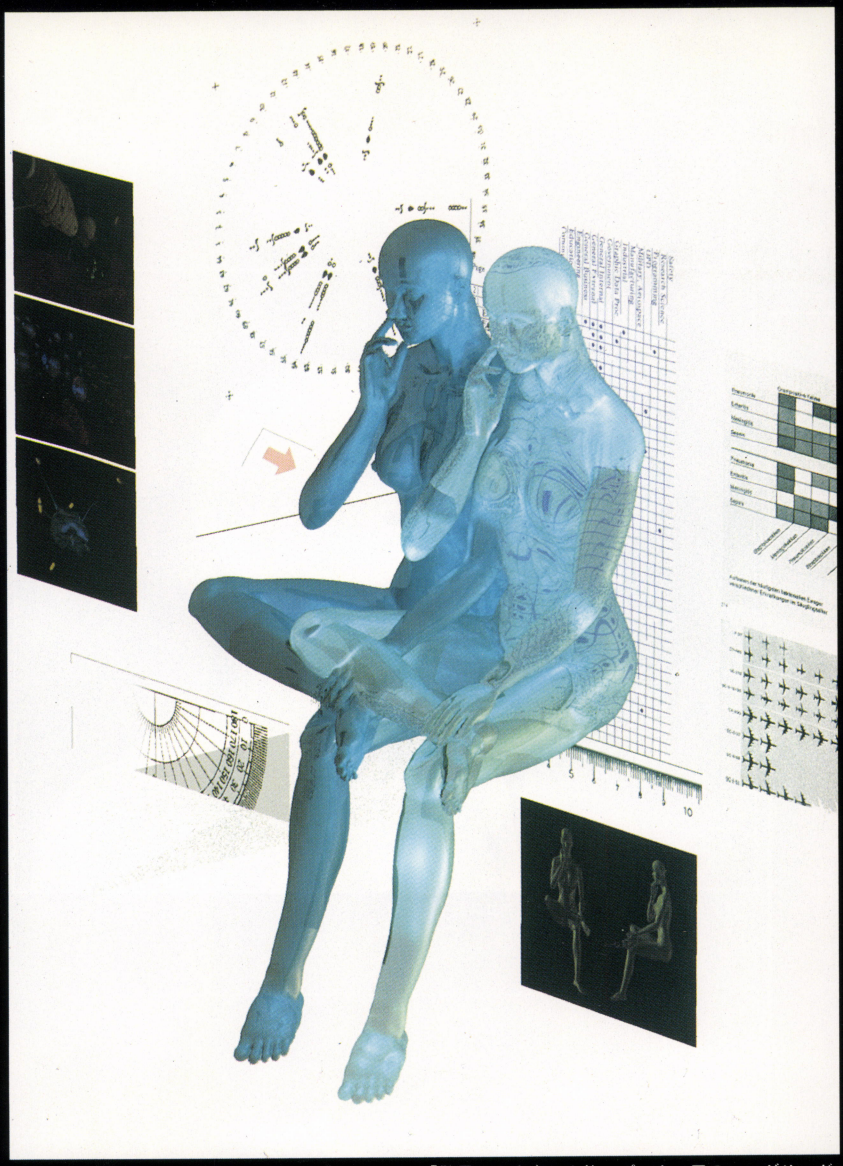
お問合わせ・作品送付先●図形処理情報センター PIXEL CGグランプリ事務局
〒101 東京都千代田区神田神保町1-64 神保町協和ビル6F ☎03(293)6161 FAX 03(293)6164

メタボール実験プログラム

日本放送協会
人体プロジェクトディレクター
伊藤 博文

本文 pp. 81〜85

実験的にメタボール（楕円メタ）を表示するプログラムを紹介する。



「驚異の小宇宙・人体」ポスター用のレンダリング

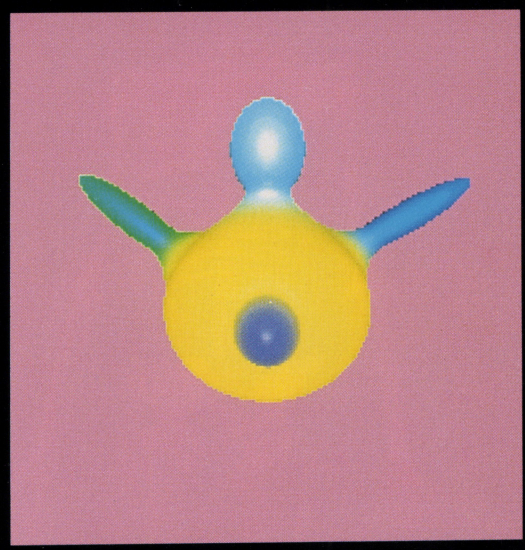


図1　サンプルデータをリスト1のプログラムによってレンダリングした結果

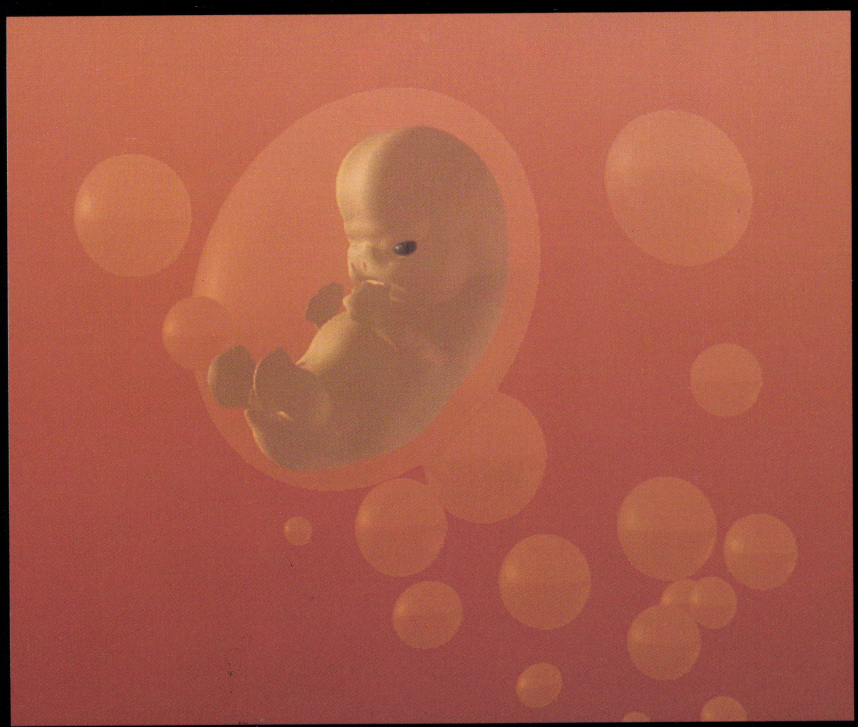


図2　金子徳明氏によるモデリングで tracygt によるレンダリング結果

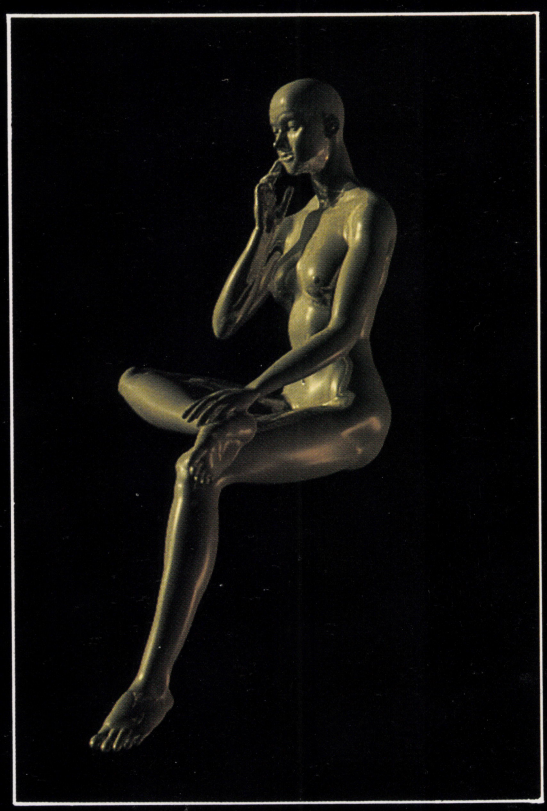


図3　金子徳明氏によるモデリングで tracygt によるレンダリング結果

メタボールと私

本文 pp. 76〜80
㈱メタ・コーポレーション・ジャパン　代表取締役　　高沖　英二

META EDITOR 開発の経緯と筆者が使用してきたシステムの変遷，メタボールの造形的特性と医学分野などへの応用例と可能性について述べる。

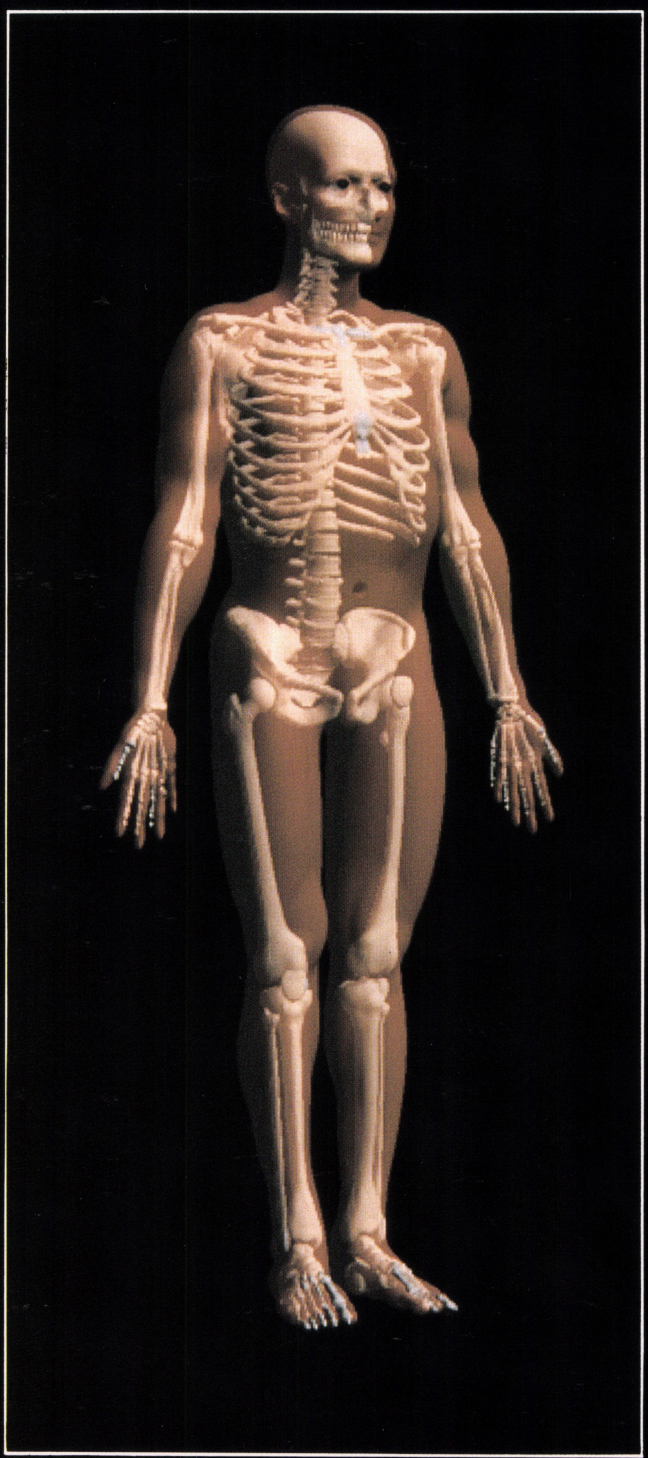


図 4　全身骨格

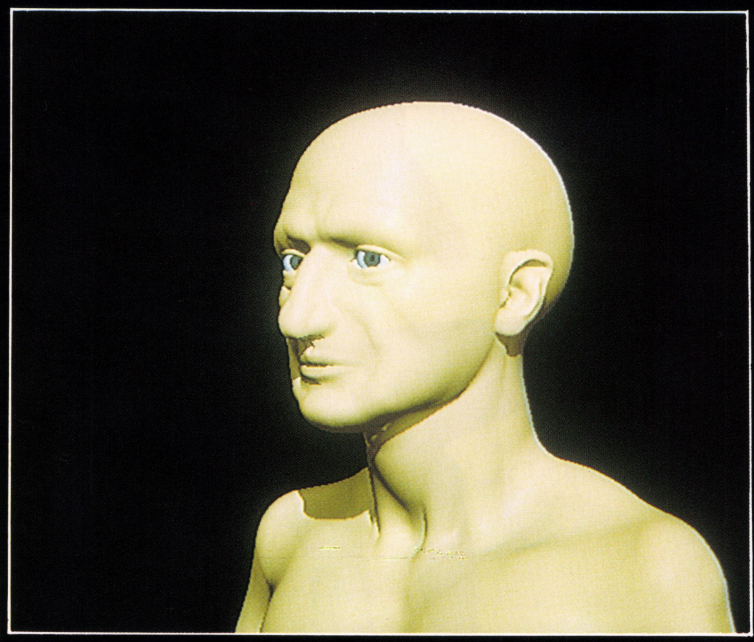


図 1　パソコンでつくったメタコス氏

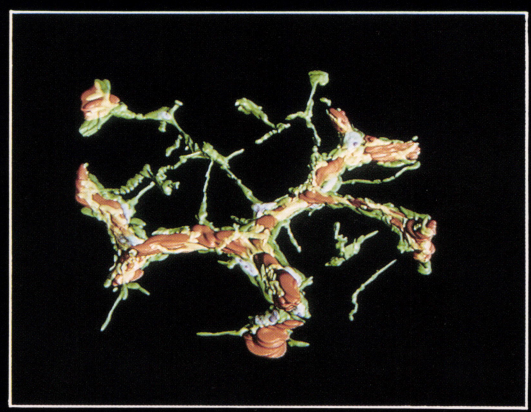


図 6　伊東細胞

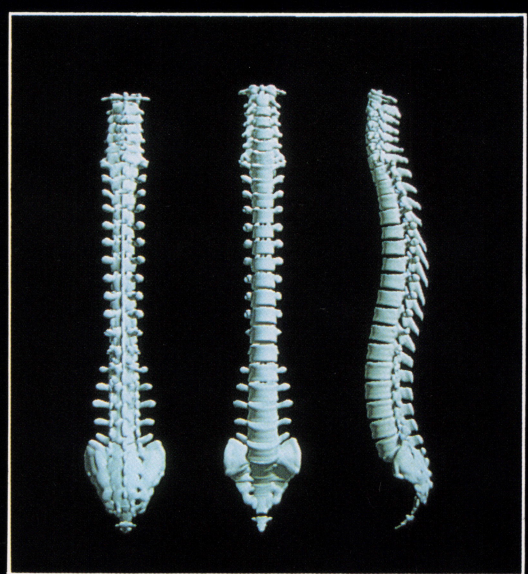


図 5　骨格部分（背髄）

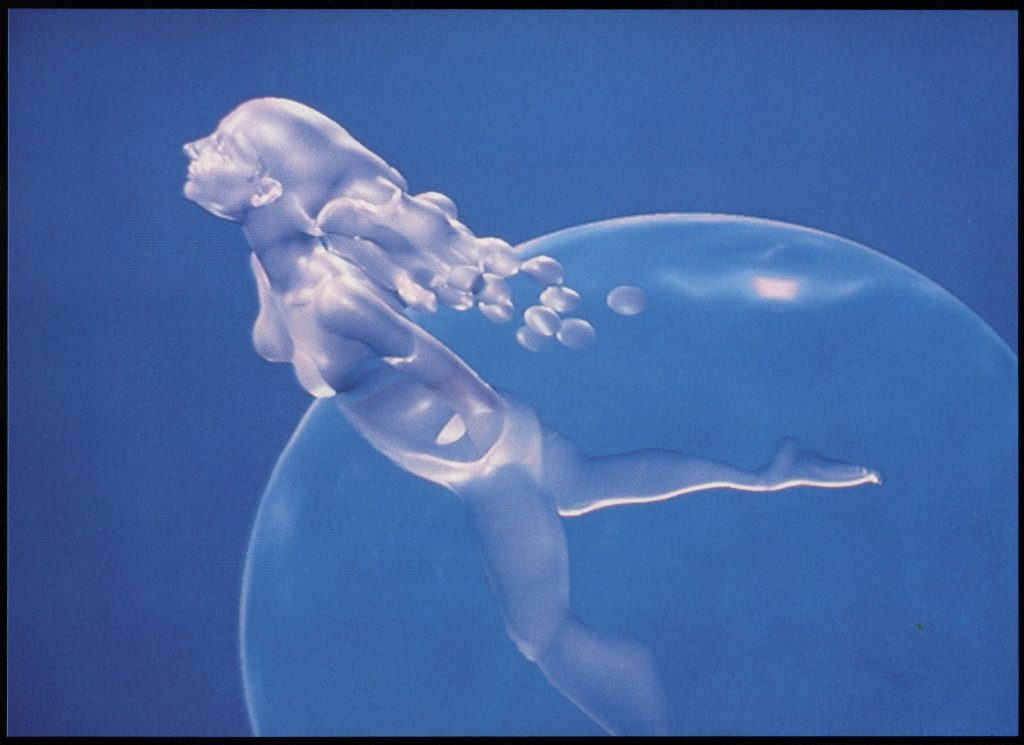


図 7

図 8

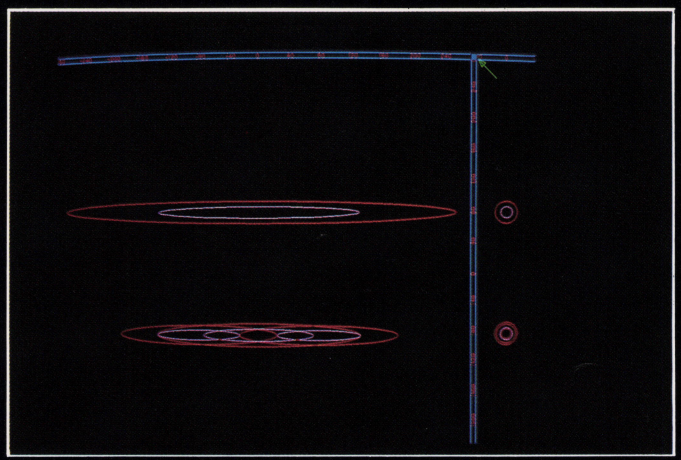


図 2　META EDITOR 画面

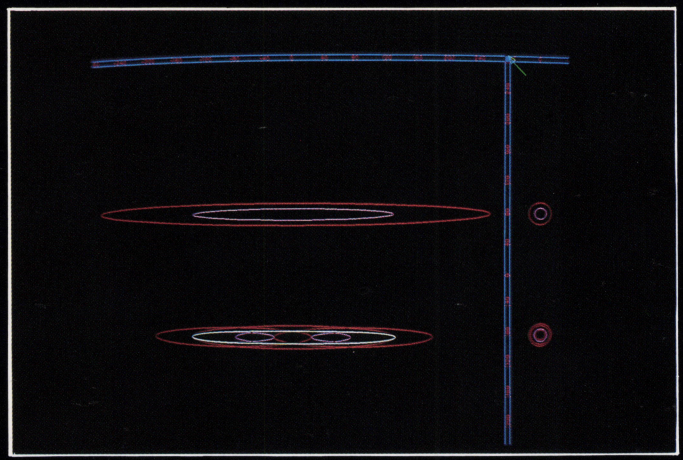


図 3　META EDITOR 画面

ここまできたリアリズム
——四半世紀の進展——

本文 pp. 133～142

広島大学工学部　　中前 栄八郎

CG が誕生してから四半世紀，リアリズムの進歩について，
SIGGRAPH を中心にしてその動向を探ぐる。

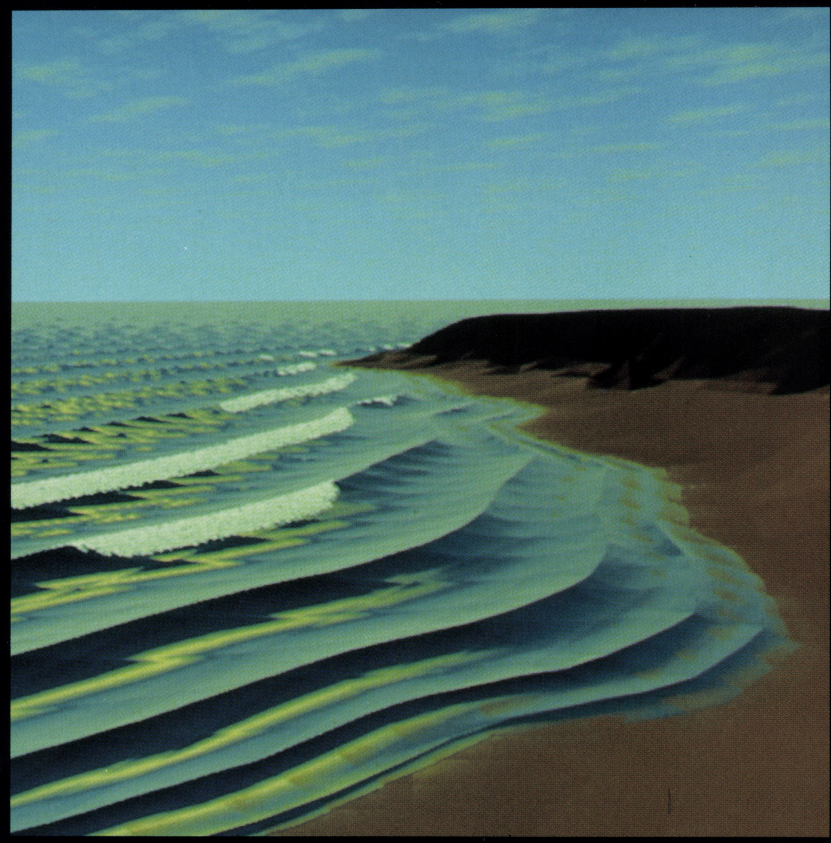


図 18　波の動きの表現

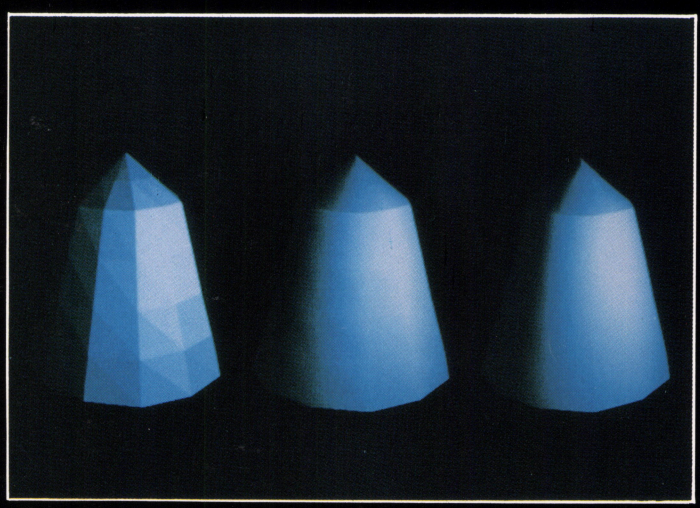


図 3　曲面の表示法

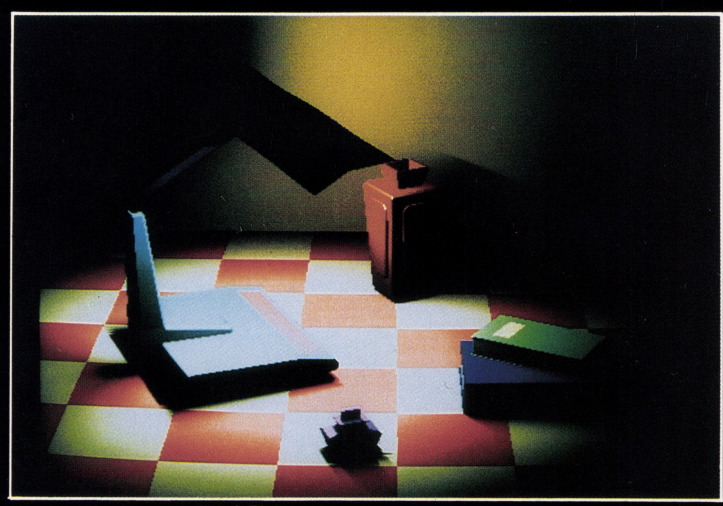
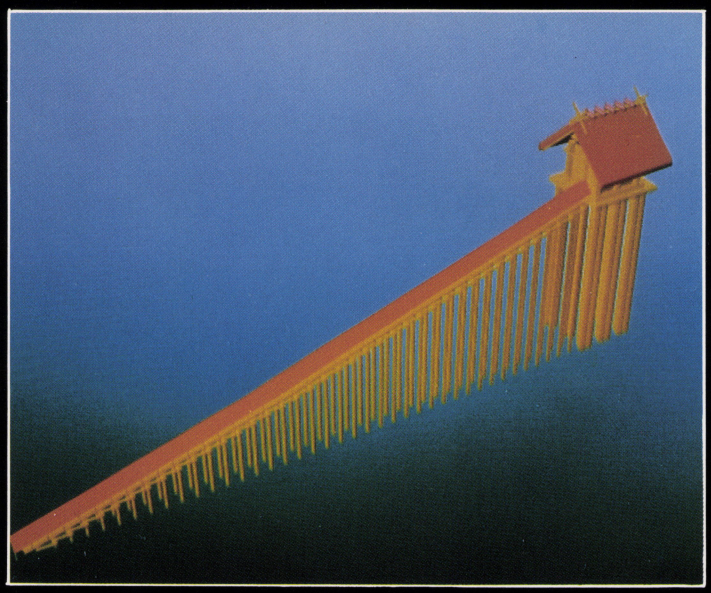
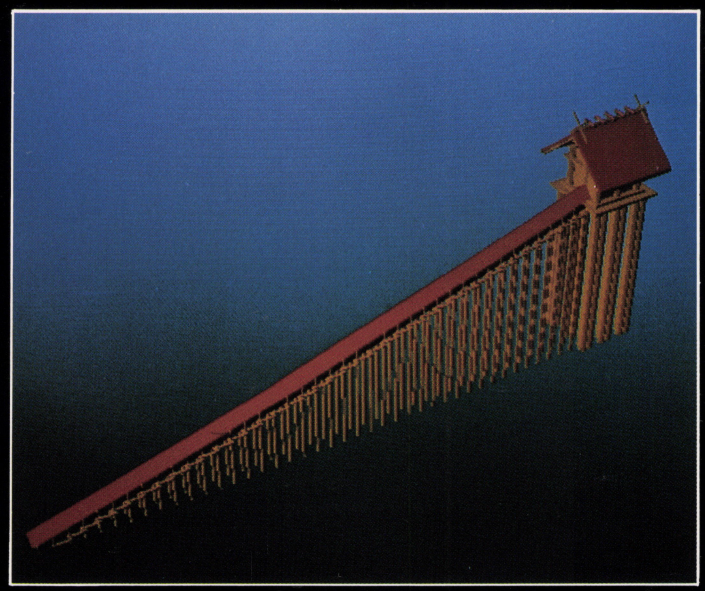


図 13　線光源による半影と本影

図 16　モーション・ブラ



エイリアシング除去



エイリアシングあり

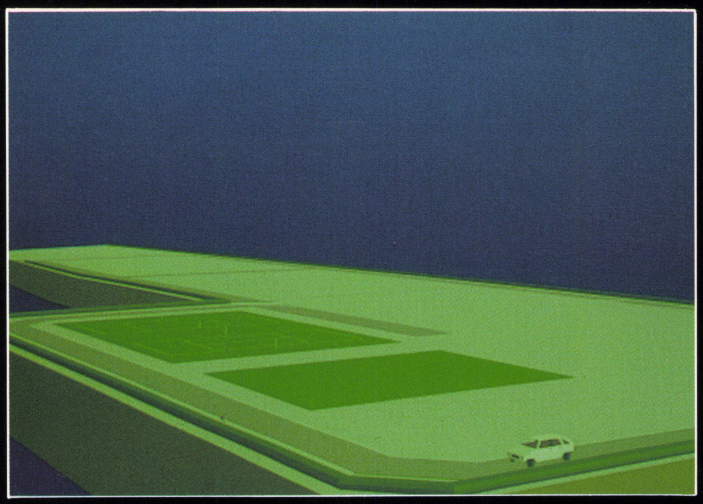
図 7　エイリアシングの除去

図 21　指数関数を用いた遠行法と投光器で照らされたビル

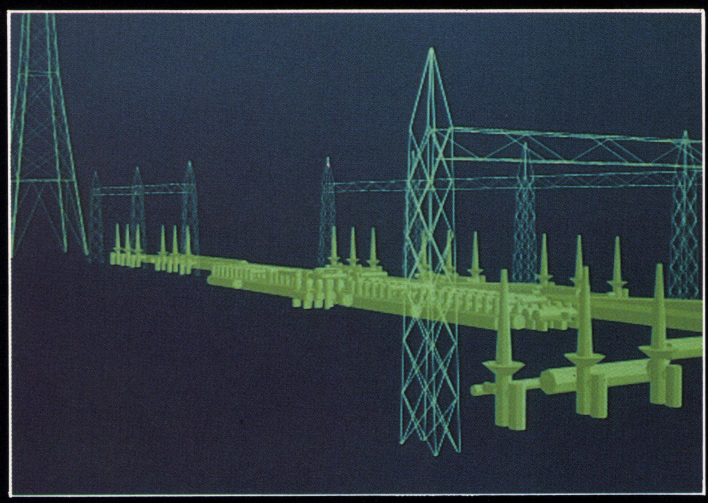
フェイズ・ファンクションなし

フェイズ・ファンクションあり

図 22　ヘッドライトで照らされた霧の表現



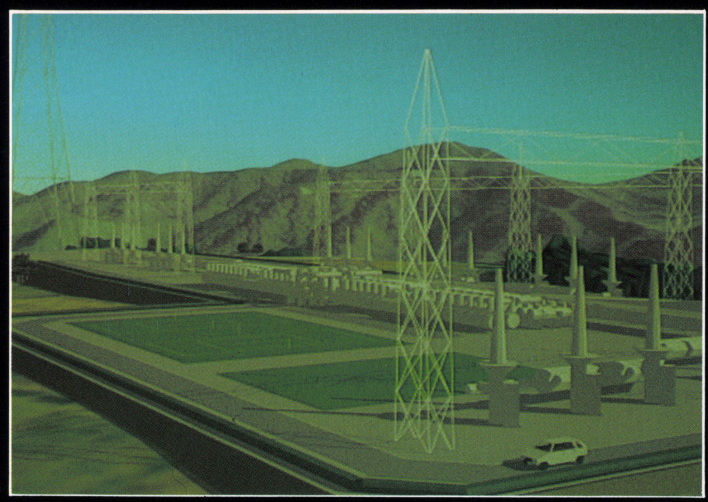
土台



電力設備

図 15　画像合成の例

背景



合成後

3次元設計用CAD概論

本文 pp. 86〜93

日立造船情報システム㈱　山本　理

設計を支援する3次元CADの機能と，それを支えるソフトウエア とリードウエアの要素技術について述べる。

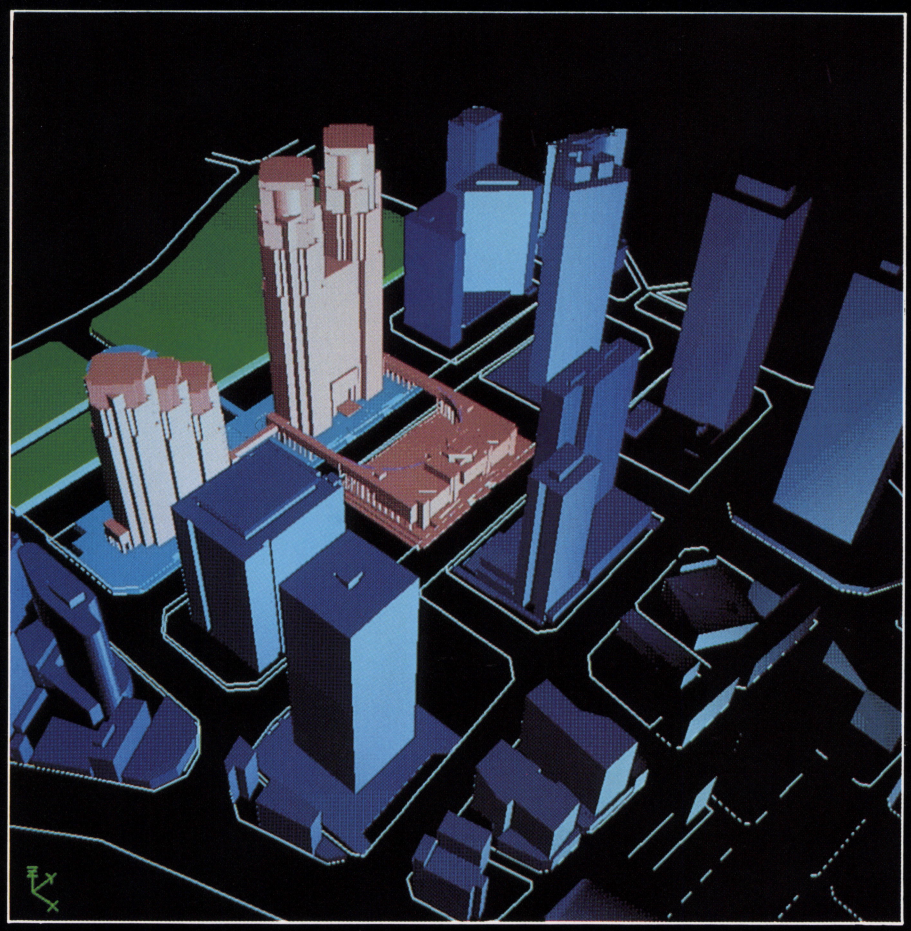


図 3

図 2

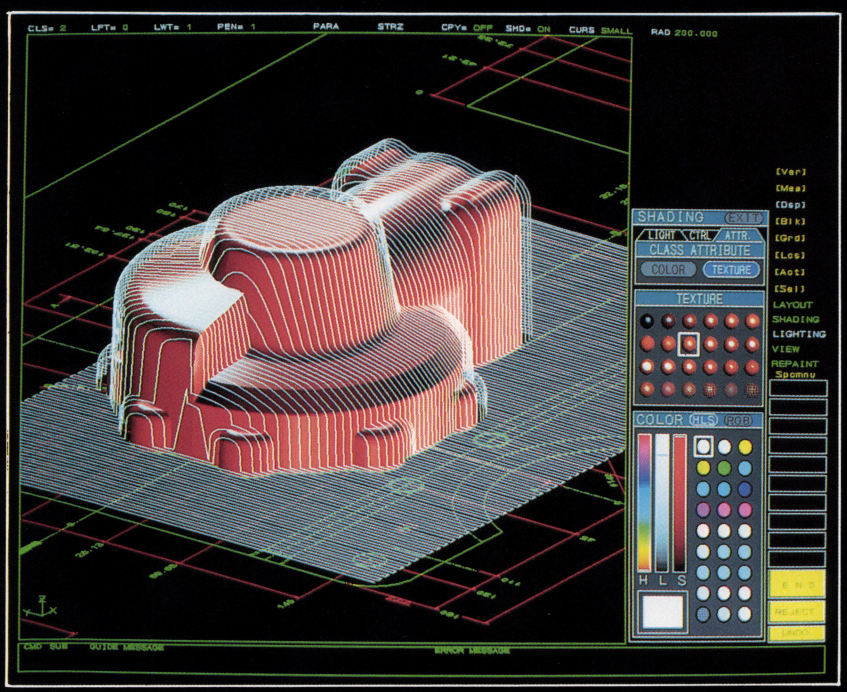


図 1

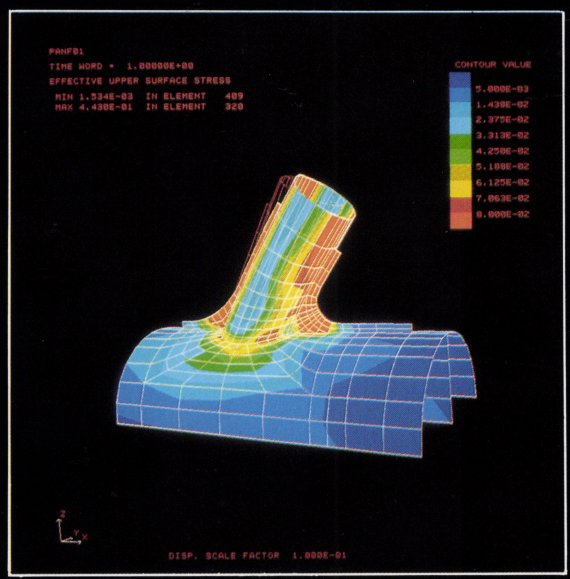


図 4

注目の新製品 3 次元レンダリングを高速処理するディスプレイ
COMTEC3555 レンダリング・ターボ

ダイキン工業㈱　電子機器事業部企画課　　増田　孝
本文 pp. 176〜178

テクスチャ＆バンプ・マッピングをはじめ，従来ソフトウエアで処理していた高度なレンダリング機能をハードウエア化した 3 次元 CG の概要と，特長について述べる。

図 12　カーテン

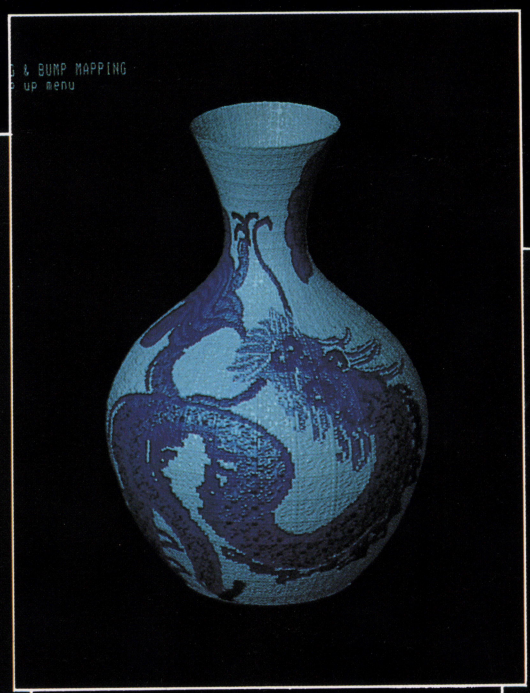


図 4　テクスチャマッピング

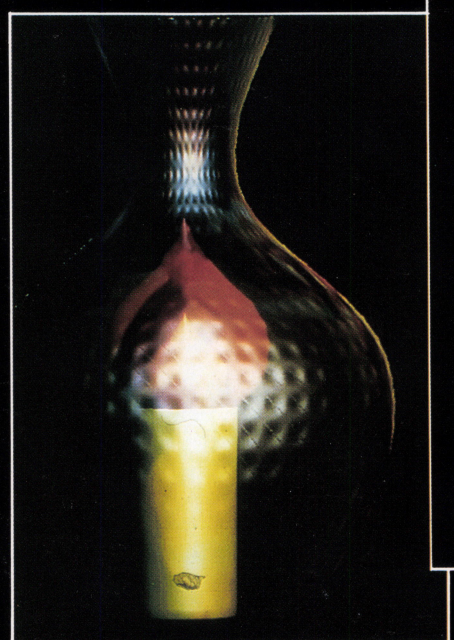


図 5　テクスチャ＋バンプ・マッピング＋透明感

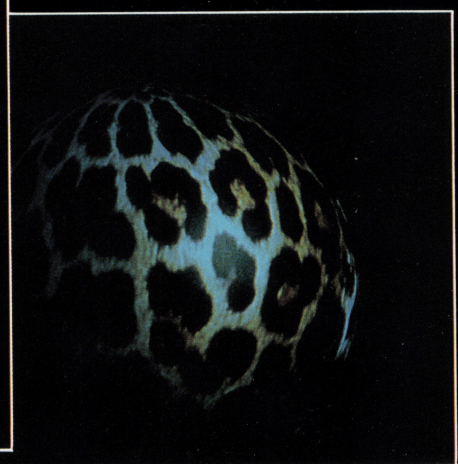


図 3　テクスチャマッピング

ボリューム・レンダリング

柴本　猛*

ボリューム・レンダリングに関して，概説，データ構造，特徴，表示方法について述べる。

ボリューム・レンダリングとは

ボリューム・レンダリングとは，ボリュームを直接レンダリングする技術である。“ボリューム”とは「3次元空間に存在する立体の，表面だけでなく内部を含む一塊り」をいう。“直接”とは，立体の内部を面などの幾何プリミティブで近似する処理なしでそのまま表示することを意味する。

従来のコンピュータ・グラフィックス（CG）のレンダリングは物体の表面を対象にしているものが多かった。立体の断面をCGで表現する場合は，幾何モデラーなどを用いて新たにその部分に面を生成しなければならない。しかし，最近CGの適用分野が拡大してきており，医療分野，結晶の分子解析，あるいは応力解析など，もともと人間にはそのままの形で見ることができないデータの可視化という方向にも範囲が広がってきた。このような分野では，測定した立体的なデータから画像を作らなければならない。従来のレンダリング技術を利用するには，測定データからレンダリングのためのデータに変換する作業が必要となる。つまり，面データを生成しなければならない。しかし，これは多くの場合うまくいかない。面を生成することは，換言すれば物体の内部と外部の境界を定義することに他ならない。しかし，たとえ測定したデータがはっきりと二値に区別できたとしても，そこから面を求めるのは形状が複雑になると大変困難である。例えば医療データの場合，対象とする物体は何種類もの組織で構成されており，おのおのの境界面を求めなければならないのでさらに複雑になる。そのうえ，実際のデータの場合，境界付近では組織ごとの値がぼんやりとしか変化しないので境界面を規定しにくい。結晶の場合，境界面はあくまで人間にとって理解を容易にする仮想的なものであり，細かい面が求めにくい。しかも，測定データにエラーがあればアルゴリズム的に正しい算出方法であっても面が求まらない。また，この測定データは面を生成するにはデータの量が少ないのが普通である。これらの問題を解決する方法として，3次元のボリュームデータを直接レンダリングする技術が要求されてきた。これがボリューム・レンダリング技術である。

ボリューム・レンダリングのデータ構造

ボリューム・レンダリングでは表面だけでなく，内部の情報ももった立体の表示を目的にしている。このような立体を定義するために，通常の場合3次元座標上に規則正しく格子状に並んだ点を使用する。これは別の見方をすれば，3次元空間に充塡している細かい立方体を考え，それらが値をもっていると考えることができる。この立方体をボクセルという。ボクセルがもつ値は色であったり密度であったりする。このようなスカラー量だけでなく，流体の解析などではボクセルがベクトルデータをもつこともある。さらに，最近ではさまざまな属性をもつようになってきた。これらのボクセルのデータを用いてレンダリングを行うが，1ボクセル8ビットを使った場合，1辺を128分割するとデータ量は2Mバイト，1辺が256の場合16Mバイトの容量になってしまうので，あまり多くの点に対してデータを得ることはできない。このため，少ないデータでわかりやすい画像を生成する技術も大切である。

データ量が多いので，これらのボクセルデータを人間が直接入力することは稀である。生物のデータであれば，CT（Computed Tomography）スキャナを用いて立体の断面データを輪切り状に何枚も測定する。流体や構造解析などの解析データの場合，測定の結果を計算機で解析し各格子点のデータを生成する。

ボリューム・レンダリングは3次元画像処理に近い

2次元のイメージデータと描画データについて考えてみる。イメージデータとはカメラやスキャナなどから得た画像データであり，データはピクセルとよばれる正方形（あるいは長方形）

*しばもと　たけし　　日本ビクター㈱　総合技術研究所　〒239　神奈川県横須賀市神明町58-7

の画素を単位として記録される。このイメージデータは直接表示できるかたちになっているが，必要であれば色の変更（特定の色の抽出），微分（エッジ抽出），フィルタリングなどの処理を行うことができる。これは簡単な画像処理である。それに対して描画データは直線・円弧などの描画操作コマンドでデータを表現したものである。データの量そのものがイメージデータに比べて著しく減少するだけでなく，1つ1つのデータを後から編集可能である。すなわち，位置を変更したり，拡大・縮小を自由に行うことができる。また，データに幾何以外の属性情報を付加することにより，データに意味をもたせることが可能である。これはグラフィックス処理である。

このイメージデータと描画データの関係を3次元に拡張するとボリュームデータと従来のサーフィス・データとの関係に対応付けることができる。つまりボリュームデータはコマンドではなく，3次元空間のその位置での物体の実データなのである。仮にボリュームデータを3次元のイメージデータと考えると，ボリューム・レンダリングは3次元のイメージデータを2次元のイメージデータに変換する処理ということもできる。もちろん，サーフィス・レンダリングと同様のシェーディング手法なども用いるが，そこで行われる処理はむしろ画像処理によったものになる。

表示方法

■従来までの表示法

ボクセル表現で3次元空間データが求まっているとする。このデータを表示する方法は従来まで大きく分けて2つあった。一つはクロスセクション・レンダリング（切断面表示法）であり，もう一つはスレッシュホールド・レンダリング（閾値面表示法）である。

(1) クロスセクション・レンダリング(切断面表示法)

最も簡単な表示方法である。xy 面，yz 面，zx 面に平行な任意の面で3次元データを切断し，切断面のデータのみを表示する。表示に際しては，2次元の切断面データを単純に表示するのではなく，切断によって切り出された直方体を3次元的に表示するのが普通である。そのために，求めた切断面のデータをテクスチャ・マッピングの要領で直方体の各面にカラー・マッピングする。

この方法は処理が簡単で単一の方法で例外なく表示できるため，ハードウエア化された例もある。欠点は，切断面だけしか表示されず，内部が見えないことである。そのため，人間が切断する位置をインタラクティブに変えながら，3次元画像を自分の頭の中で再構築しなければならない。ハードウエアで高速表示することはインタラクティビティ向上のために必要である。また，任意の角度で切断することもできると思うが，実現例はないようである。

(2) スレッシュホールド・レンダリング(閾値面表示法)

3次元データ中に存在する複数の物質の中から観察の対象とする特定の物質のみを表示する手法である。基本的には，その物質に対応する値をもったボクセルをつないで面を作り従来の手法で表示するのだが，表示品質を上げるためにさまざまな方法が提案されてきている。処理はおおむね3段階に分けられる。第1は同一の値（あるいは一定の範囲にある値）をもったボクセルを探して近接したものを接続する処理，第2は接続されてできる形状を面で再構成（近似）する処理，第3は面にシェーディングを施し表示する処理である。

この表示法は医療用の応用が多く，人体の表示などに用いられてきた。人体などの場合，骨と肉のようにはっきりと分かれた物質で構成されており，データ値の差異が明確でデータの連続性もある。そのため，多面体や曲面による近似が行いやすい。CT スキャンや MR 画像によるデータを使うことが多いが，スライスされたデータから正確に接続関係を見つけ，面を再構成することはかなりの困難を伴う。

以前試みられていたのは，1枚のスライス画像の中で輪郭抽出を行い，得られた輪郭線をスライス順に接続する方法であった。通常，インタラクティブに輪郭抽出を行って自動的に輪郭線をつなぐ場合が多いが，接続ミスが起こりやすい。

ここでは，優れた方法として「Marching Cubes Algorithm」

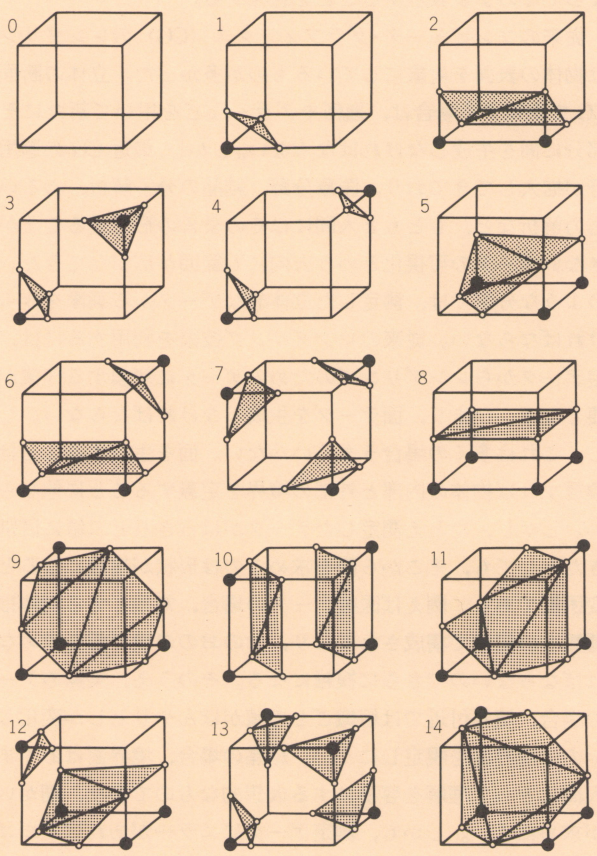


図2　面の貼り方の種類(COMPUTER GRAPHICS, Vol. 21, No. 4, pp. 165)

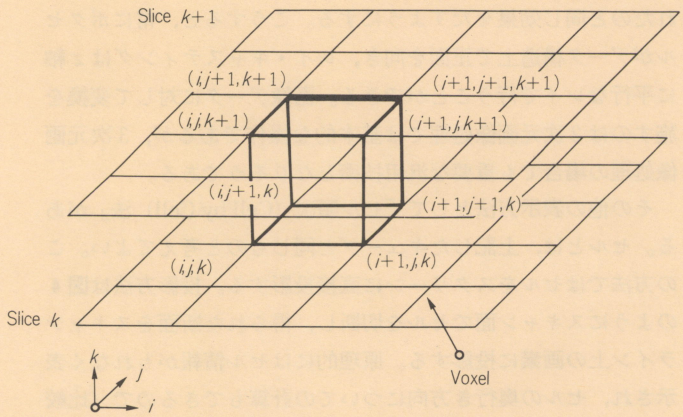


図1　Marching Cube(COMPUTER GRAPHICS, Vol. 21, No. 4, pp. 164)

を取り上げて説明する。この方法では8個のボクセルを頂点とする立方体（キューブ）を考え，あらかじめ指定した値(＝表示したい面がもつべき値)と各ボクセルがもっている値との大小関係を調べ，表示したい面がキューブのどの位置にあるかを判定し，キューブの中に面を再構成していく。これは以下のようにして実現される。

キューブは隣接した8個のボクセルを頂点として，**図1**のように定義される。キューブの頂点の値が表示したい面の値より大きいか等しいときには頂点に1を立てる。1が立った頂点は面の内側または面の上にあると考える。頂点の値が表示したい面の値より小さいときには頂点に0を立てる。0が立った頂点は面の外側にあることになる。この頂点の1か0かの値を使うと，キューブのエッジを面が交差するかどうかを判定できる。面は3角形で考える。この交差の仕方は$2^8=256$通りあるが，1と0が逆の関係のものを同じトポロジと考え，回転対称を考えると**図2**の14種類に減らすことができる。

キューブのエッジと3角形の交差位置（＝3角形の頂点座標）は，エッジの両端点のデータ値と求める面の値(＝3角形の頂点がもつ値）から比例配分で求める。

3角形頂点のノーマルベクトルを求めるには，キューブの頂点のノーマルベクトルを求め，2頂点のノーマルベクトルを内挿して3角形頂点の値とする。キューブの頂点のノーマルベクトルはキューブの頂点でのデータ値の勾配ベクトルと同じと考える。キューブの頂点でのデータ値の勾配ベクトル $\boldsymbol{G}$ の3成分は隣接キューブの頂点データ値 $\boldsymbol{D}$ を使って，

$$\boldsymbol{G}_x(i,j,k)=(\boldsymbol{D}(i+1,j,k)-\boldsymbol{D}(i-1,j,k))/\Delta x$$
$$\boldsymbol{G}_y(i,j,k)=(\boldsymbol{D}(i,j+1,k)-\boldsymbol{D}(i,j-1,k))/\Delta y$$
$$\boldsymbol{G}_z(i,j,k)=(\boldsymbol{D}(i,j,k+1)-\boldsymbol{D}(i,j,k-1))/\Delta z$$

と表される。ただし，Δx，Δy，Δz はキューブのエッジの長さである。これらを正規化するとノーマルベクトルとなる。データ値の勾配ベクトルが面のノーマルベクトルと同じと考えてよいのは，表示したい面の接線方向のデータ勾配は0であり，データ勾配ベクトルは求める面に垂直な方向（＝ノーマルベクトル）と考えられるからである。

最後に，求まったノーマルベクトルを使ってシェーディングを行う。

■ボリューム・レンダリング

ボリューム・レンダリングは面で近似することなく，ボクセ

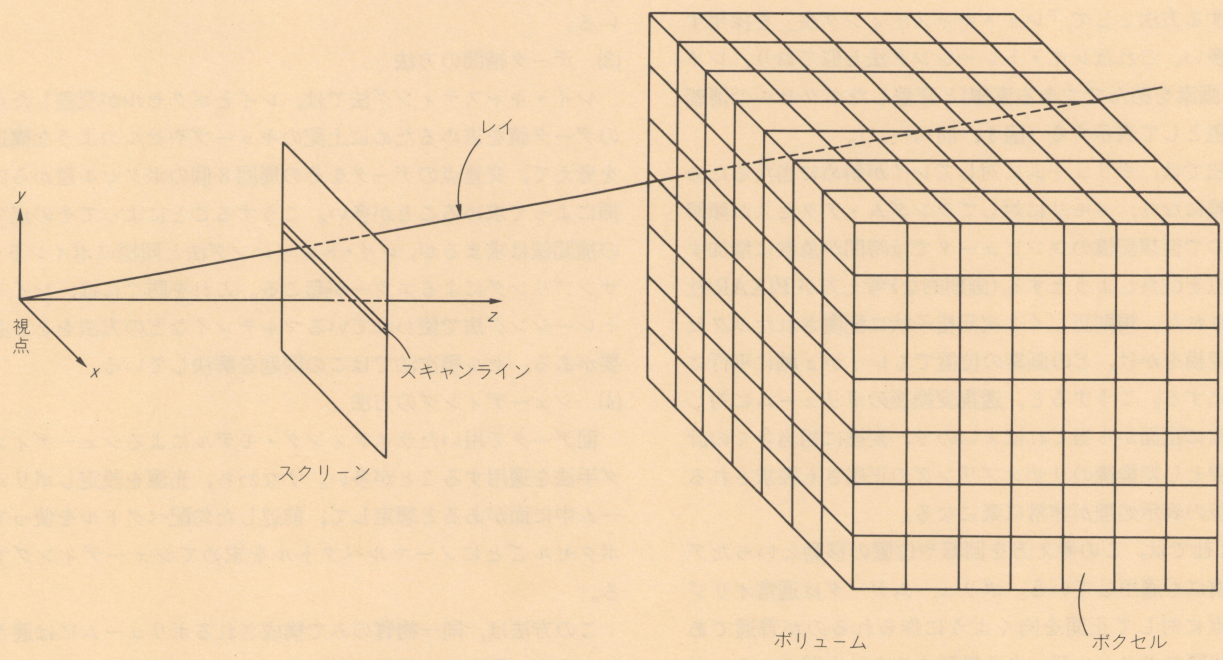


図3　レイ・キャスティング法

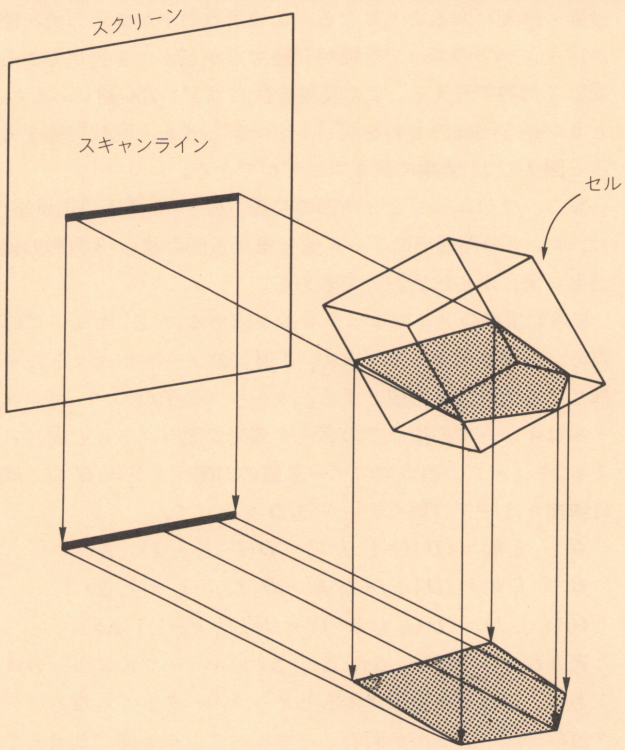


図4　セル順次法におけるセル・スパンの求め方

ルデータ値を直接表示し，内部の様子も観察できるようにする。ここで主要な技術は，ボクセルデータをどのようにしてスクリーン上に投影するか，内部をどのようにして見せるか，データ補間とアンチ・エイリアシングをどのように行うか，シェーディングをどう行うかなどである。

(1)　スクリーンへの投影方法

ボリューム・レンダリングでは，ボクセルデータをスクリーンへ投影する方法として「レイ・キャスティング法」を採用することが多い。これはレイ・トレーシング法と似ており，レイ(＝視点と画素を結んでできる直線)と交差したボクセルの情報を画素の色として表示する（**図3**，前ページ)。

この方法では，ボリュームに対してレイが斜めに当たるため計算が複雑になり，メモリに対してランダム・アクセスが頻繁に起こるので仮想記憶のコンピュータでは時間が急激に増加する。この点を改良しようとする(画期的な)考え方がPIXAR社から提案された。規則正しく3次元格子状に配置されたボクセルに透視変換をかけ，どの画素の位置でもレイがz軸に平行になるようにする。こうすると，透視変換後のボリュームに対してレイは常に正面から当てればよいので，変換に相当多くの計算量を必要とし変換後のリサンプリングの正確さも要求されるが，その後の表示処理が非常に楽になる。

PIXAR社では，この考え方を回転や位置の移動といったアフィン変換にも適用している。ボリュームデータは通常オリジナルの視点に対して正面を向くように作られるのが普通であり，この状態でボクセルデータを投影するのは上記のようにすれば簡単にできる。しかし視点が変換を受けた場合，ボクセルに対してレイが斜めに当たってしまう。そこで，視点の変換に際して視点を変換せずボクセルデータを逆変換し，視点が変換されたのと同じ効果をだすようにする。こうすると，常にボクセルがデータ構造上で正面を向き，レイ・キャスティングはz軸に平行なレイで行うことができる。画像データに対して変換を施すのは2次元画像処理では基本的な操作であるが，3次元画像処理の場合でも重要な汎用技術となりそうである。

その他の表示方法として「セル順次（Cell by Cell）法」がある。セルとは，上記したキューブと同じものと考えてよい。この方法ではセルをスクリーンに直接投影する。投影方法は**図4**のようにスキャン面でセルを切断し，得られた断面をスキャンライン上の画素に投影する。原理的にはセル情報がもれなく表示され，セルの奥行き方向についての計算もできるので（比較的少ない数のセルでボリュームを表したときに）レイ・キャスティング法がもっているポイント・サンプリングの欠点を解消するのに有効である。

(2)　内部の見せ方

ボリュームの内部を見せるために半透明表示手法を使うのが普通である。この場合，もともとのボクセルデータを寄与率(不透明度)αとして使うこともあるが，新たにα値を別に設けると，目的とする物質のα値を他の物質よりも大きくして見やすくすることもできる。レイ・キャスティングで表示する際，手前のボクセルから順番にボクセルのデータ値をα倍$(0 \leqq \alpha \leqq 1)$して加算していき，αも加算していく。αの総和が1となるか，レイがボリュームを通過したら1つの画素への投影を終了し，加算したデータ値をその画素の値として表示する。

上記の方法は，ボリュームに物質が1種類だけしか存在せず密度のみが異なるような場合，正確さを欠いた表示となる。このような場合のために光の伝播を考慮したモデルも考えられている。

(3)　データ補間の方法

レイ・キャスティング法では，レイとボクセルが交差した点のデータ値を求めるために上記のキューブやセルのような構造を考えて，交差点のデータをその周囲8個のボクセル値から内挿によって求めることが多い。こうすることによってその点での補間値は求まるが，レイ・トレーシング法と同様にポイント・サンプリングによるエラーが起こる。これを防ぐには，レイ・トレーシング法で使われているマルチレイなどの方法をとる必要がある。セル順次法ではこの問題を解決している。

(4)　シェーディングの方法

面データで用いたライティング・モデルによるシェーディング手法を適用することが多い。すなわち，光源を設定しボリューム中に面があると想定して，前述した勾配ベクトルを使ってボクセルごとにノーマルベクトルを求めてシェーディングする。

この方法は，同一物質のみで構成されるボリュームには適さないので，発光する粒子の集まりによる光の伝わり方をモデル化することもある。

物理法則ベースモデリング

寺沢　幹雄*　　柴本　猛**

コンピュータ・アニメーションの分野で最近話題の物理法則ベースモデリングについて概説し，その研究動向をサーベイする。

物理法則ベースモデリングとは

1987年ころから物理法則ベースモデリング（Physically-Based Modeling）という新しい言葉が，主にコンピュータ・アニメーションの分野で広く使われるようになってきた[4]。これは，コンピュータ・グラフィックスの映像をより現実世界に近付けるために，自然界を支配する諸法則（慣性則，運動量保存則など）をコンピュータの世界にも取り入れたものである。アニメーションの制御性の良さに，科学シミュレーションの正確さを取り入れるためのモデル化といってもよい。しかし，語の定義は必ずしも明確ではなく，人によっては雲や樹木の表示のように広く自然現像のモデル化ととらえる場合もあれば，狭くリンク機構の動力学を用いたモデル化ととらえる場合もある。ここでは，物理法則ベースモデリングを「コンピュータ・アニメーションにおいて，自然な動きを実現するために自然界の物理法則を用いて物体や現像を実用的な時間で計算する汎用な手法」と考えることにする。

歴史的背景

これまでに，コンピュータ・アニメーションにおいて動きを指定するための方法がいくつも提案されてきた。これを大きく分類すると次のようになる[31]。

1)　対話法（Interactive Method）

ダイヤルやジョイスティックなどの装置を使ってリアルタイムに動きを確認し，試行錯誤で動きを決定していく指定方法。あるフレームでの位置や姿勢を指定し，間のフレームを補うキーフレーム補間，対話的に物体の軌道を生成する軌道指定法，各自由度に対して制御関数を指定する制御関数法が対話法としてあげられる。

2)　スクリプト・アニメーション（Scripted Animation）

個々の自由度の指定ではなく，マクロな動きを台本（Script）で記述する指定方法。アニメーション記述言語，ロボット言語などの言語による指定がこの例としてあげられる。

また，1)，2)の中でも，コマンド・ライブラリ，有限状態マシン，階層化，パラメータ化といった高レベルのものと，軌道や関節の動きを各自由度に対して指定する低レベルなものがある。

こうした従来の方法で実現できる動きには限界がある。例えば，人のように複雑な動きを表現する場合，多くの指定を慎重に行わなければ自然な動きを得るのは難しい。しかし，よく考えてみるとそれらの労力の多くは，物理法則を満たすのに費やされている。歩く動作を考えてみると，足は急に持ち上がらない，脚が曲がったときには筋肉が膨れる，着地時に床の下に足が突き抜けない，着地の衝撃は足の先から伝わる，また床の滑り具合いや歩く速さによっても歩行の様子は異なる…。もともとアニメーションでは，現実世界ではあり得ないようなことまで表現できる高い自由度をもつが，実際に全く現実を離れた動きが必要になる場合はそう多くない。このために，物理法則を考慮することはアニメーターの労力を大幅に軽減し，これまでできなかったようなリアルなアニメーションが可能になる。

もちろん，トラディショナルなコンピュータ・アニメーション手法においても物理法則を全く考慮してこなかったわけではなく，運動学的（Kinematic）には考えられてきた。運動学は速度，加速度のような時間と位置の関係のみであり，力，トルク，相互作用などには関係しない。しかし，最近の物理法則ベースモデリングでは，動力学（Dynamics），弾性学（Elasticity Theory），レオロジー（Rheology）などのより高度な手法を用いている。これらの手法自体は新しいものではなく，何十年，何百年と研究され，確立されてきた分野である。それが近年になってコンピュータ・アニメーションの分野で急に使われ始めたのは，最近のコンピュータ（特にパーソナルコンピュータ，ワークステーション）の急速なパフォーマンスの向上によるところが大きい。つまり，これまで大型汎用計算機でしか実現できないと考えられていたような高度な計算が，廉価な計算機で実現可能になってきたためである。また，もう一つの理由としてロボティックスでの研究成果があげられる。ロボティックスでは，精度向上や剛性計算のために多関節型のロボットの制御に動力学的解析（Dynamic Analysis）を適用し，リアルタイムに計算できるようなアルゴリズムを工夫した。これが，人体などのような関節でつながる物体のアニメーションに適用されるようになってきた。さらに，コンピュータ・ビジョンでの研究成果も柔

*てらさわ　みきお　　**しばもと　たけし　　日本ビクター㈱　総合技術研究所　〒239　神奈川県横須賀市神明町58-7

軟物体のモデリングに影響を与えている。

拘束ベースモデリングと目的指向モデリング

物理法則ベースモデリングと密接なかかわりをもつものに，拘束ベースモデリングと目的指向モデリングがある。

一般に物理法則ベースモデリングでは，物体がどのように振る舞うかは予測できない。例えば，ボールに位置と初速を与えたとき，ボールがどこに飛んでいくかがわからない。アニメーションではアニメーターの意図した動きを伝えることが大切なので，途中の動きをもう少し細かく制御する必要がある。このためにいくつかの拘束を定義し，制御しようとするのが拘束ベースモデリング（Constraint-Based Modeling）である。拘束にはさまざまなものが提案されている。例えば，Witkin らは物理的でないエネルギー関数を用いて点の固定，面と面の接触，滑り対偶などの拘束を定義している[34]。従来のキーフレーム法や軌道生成法なども，物体の動きに拘束を加えているという意味で拘束ベースモデリングのサブセットと考えられる。より一般的に拘束条件を機構設計に応用していこうという研究に文献6),7),37)がある。

物理法則ベースモデリングと拘束ベースモデリングは相いれない概念ではなく，後者の手法で制御し，前者の手法で自然な動きを計算するといったことが可能である。例えば，Isaacs らは慣性系を考慮した振舞い関数で拘束を定義して人の動きを求めたし[19]，Witkin らは時空間法を用い，簡単な運動学的拘束の指定により関節物体であるランプの跳躍をリアルに見せるアニメーションを作成した[35]。

Barr は，物理法則ベースモデリングや拘束ベースモデリングを含んだかたちの，物体の統一的な表現の枠組みを目的指向モデリング（Teleological Modeling）とよんでいる[5]。この枠組みでは，目的とする動きを2次元表示画面上のイメージとして得るために対象を4層に階層化し，概念を整理している。

各階層の対象となる要素は，①位置，方向，力などの拘束関係といった目的イベント要素の時系列，②時間によらない剛体，柔軟物体といったニュートン力学的要素，③多面体，パッチといった3次元形状要素，④画素の色や線素といった2次元イメージ要素である。①～③の各階層において，それぞれ目的論的手法，数学的シミュレーション，レンダリングという方法をとることにより次の階層での要素を得られ，最終的な2次元表示画面上のイメージが求まる。

物理法則ベースモデリング手法

最近発表された物理法則ベースモデリングの論文について以下に概説する。ここにあげた文献の中には，いくつかの分類にまたがっているものがあることをご了承願いたい。

(1) 剛体

時間が経過しても変形せず，ある時間での状態を物体座標の平行移動，回転移動だけで記述できる物体を剛体（Rigid Body）とよぶ。剛体の中で最も単純なモデルは質点である。質点の場合には，「質点の運動量の時間変化は，その質点に作用する力に等しい」[39]というニュートンの第2法則をもとに，この力として重力などの相互作用，摩擦力，粘性力，弾力などを当てはめ，運動方程式を解けばよい。ただし，慣性座標系ではなく加速度座標系の場合には，有効な力を得るために，慣性力として直進力の他に遠心力，コリオリの力，回転の加速による力を考慮しなくてはならない。Lundin は，地上を動き回るロボットの上下方向の動きに，ばね・マス系の解析を適用している[22]。いくつかの物体の集まった系（System）の場合には，エネルギー保存則，運動量保存則も考慮しなければならない。パーティクル・システム（Particle Systems）などは，火や水などを小さな質点の集合体と考え，問題の単純なモデル化を図っている[26]。一般の剛体の場合には直進運動の他に回転運動を考慮しなければならないので，問題はより複雑になる。Hahn は，一般の剛体の個々の動きや系としての相互作用について，汎用性のある定式化と解法を提案している[15]。

(2) 柔軟物体

布やゴムボールのように時間変化につれ変形するものを柔軟物体（Flexible Body）とよぶ[16]。ここでは，ねじり，引っ張り，引伸しのような連続変形の他に，引裂きのような不連続変形する物体も柔軟物体に含めて考えることにする。

Weil は布を縦糸，横糸の集まりとし，基準となる糸に懸垂線を当てはめ，糸の長さ一定の条件から布のモデル化を行った[29]。同様に布のモデル化を行ったものに文献11）がある。Terzopoulos らは，ゴム，布，紙，柔らかい金属といった，より一般的な弾性変形体についてのモデル化を行った[27]。この論文では，内力，減衰，弾性，外力の時間関数についてのラグランジェ形式の微分方程式を解いている。さらに，適応力として衝突回避力も加え，非貫通物体の扱いもモデル化している。文献28）ではこれを発展させ，粘弾性，塑性，破壊についても考慮している。Platt らは物理法則と最適化理論に基づいて拘束を導入することで，柔軟物体に対するアニメーターの制御を可能にした[25]。

(3) 関節物体

関節物体（Articulated Body）とは，リンク機構のように関節を介していくつかの要素が集まってできた物体のことである。要素自体は剛体でも柔軟物体でもかまわないが，現状では多くの場合剛体系であり，要素どうしが木構造状の親子関係をなし，親子の間は回転1自由度の関節でつながっている。人や動物やロボットといった数多くの物体が関節物体として扱えるので，コンピュータ・グラフィックスにおいて関節物体は非常に重要な対象となっている。このため，多くの研究がなされている[3),8),9),14),20)]。

関節物体の運動方程式には，ラグランジェ方程式とニュートン-オイラー方程式の2つがある[40]。

ラグランジェ方程式（Lagrangian Equation）は，運動エネ

ルギーと保存エネルギーをもとに方程式を立てており，定式化が容易で理解しやすいが，計算効率が比較的悪い。このため漸化式表現にして，リアルタイムに計算しようとする研究がなされている[17),18)]。漸化式表現では，最初に親から子へ順々に直進，回転の速度，加速度を求めていき，次に子から親へと力とトルクを計算していくという方法をとる。Wilhelmsらは，同様に，各リンクでの運動エネルギーの関係に基づいたGibbs-Appellの定式化により関節物体の動力学を解いている[30)]。

ニュートン-オイラー方程式（Newton-Euler Equation）は，各リンクに対して直進運動はニュートンの方程式，回転運動はオイラーの方程式を適用したものである。ニュートン-オイラー方程式についても漸化式表現がなされ，効率的な解法が示されている[21)]。Armstrongらは関節にトルク関数を与え，ニュートン-オイラー方程式に基づいて，直接動力学的に解くための効率的な（O(n)の）アルゴリズムを考案した[1),2)]。FeatherstoneもO(n)の効率の良い解法を示した[10)]。

関節物体に対する制御としては，力やトルクを与えて軌道を生成する直接動力学（Direct Dynamics）と，軌道から各関節の力やトルクを求める逆動力学（Inverse Dynamics）がある。モーターで動きを制御する場合には前者を，運動軌跡で拘束する場合には後者を用いる。どちらを解く場合でも，もとになる式自体は同じだが，漸化式表現にした場合，どちらの立場で解くかによってアルゴリズムが異なってくる。

(4) 衝突回避

コンピュータ・グラフィックスの世界では，物体どうしが互いにぶつかってもそのまま貫通してしまう。このような不自然な振舞いをなくすために，物体どうしの干渉を避けることを衝突回避という。大きく分けると衝突回避は，物体が剛体でありはね返ることで衝突を避けるものと，物体が柔軟物体であり物体自体が変形して衝突を避けるものがある。前者の例として，Mooreらは柔軟物体や多面体に対する衝突検出法と，そのときの衝撃による物体系の振舞いの動力学的な解析を述べている[23)]。後者の例として，寺沢らは自由形状変形を用いて衝突による物体変形をモデル化した[38)]。

(5) 自然現象

自然現象のモデル化ではなんらかのかたちで物理法則が考慮されている。

その中でも，重力や幾何学的な拘束などの影響を考慮した例として，文献12)，13)，24)などの波のモデリングがある。また，流体力学などを用いて木星の雲の流れをシミュレートした文献36)などがある。

おわりに

物理法則ベースモデリングは，米国において近年急に話題になった研究分野であり，1987年のACM SIGGRAPHでもTopics in Physically-Based Modelingというコースが新たに設けられた。また実際，従来とは比べものにならないほど動きのリアルなアニメーションが何本か作られている。これらの作品を見ると，単に現実世界の模倣だけでなく，コンピュータ・グラフィックスの表現力が格段に増しているのがわかる。今後，コマーシャル・フィルム，番組タイトルをはじめ，プレゼンテーション・グラフィックス，CAIなどさまざまな分野で物理法則ベースモデリングは強力なツールとなっていくであろう。

日本では，まだあまりこの分野に関する論文は出されていないようであるが，物理法則ベースモデリングは裾野の広い分野であり，ロボット制御をはじめとしたさまざまな基礎研究は進んでいるので，今後コンピュータ・アニメーションへの適用が期待できる。

なお，基礎的な知識の多くは大学の教養課程程度の力学で十分であり，1987年のSIGGRAPHのコースノート中の文献33)が簡潔にまとまっている。動力学的解析手法については，文献2)，32)，40)などに詳しい。その他に，材料力学，弾性力学，塑性力学，流体力学，レオロジーなどの専門書が参考になるであろう。

参考文献

1) W.W. Armstrong：“Recursive Solution to the Equations of Motion of an N-link Manipulator”, Proceedings of Fifth World Congress on Theory of Machines and Mechanisms, Montreal, 2, American Society of Mechanical Engineering, 1979

2) W.W. Armstrong, M. Green：“The Dynamics of Articulated Rigid Bodies For Purposes of Animation”,Proceedings of Graphics Interface 85, Montreal, Canada, 1985

3) N.I. Badler, J.O' Rourke, B.Kaufman：“Special Problems in Human Movement Simulation Computer Graphics, Vol.14, No.3, 1980

4) A.Barr, R.Barzel, D.Haumann, M.Kass, J.Platt, D.Terzopoulos, A. Witkin：“Topics in Physically-Based Modeling”, SIGGRAPH'87 Course Notes,17, Anaheim, Calif.,1987

5) A.Barr：“Teleological Modeling”,SIGGRAPH'88 Course Notes,27, Atlanta, Ga.,1988

6) R.Barzel.：Dynamic Costraints, MSc thesis, Department of Computer Science, California Institute of Technology, Pasadena, CA,1987

7) R.Barzel, A.H. Barr：“A Modeling System Based on Dynamic Constraints”, Computer Graphics, Vol.22, No.4, 1988

8) T.W. Calvert, J. Chapman, A. Patla：“The Integration of Subjective and Objective Data in the Animation of Human Movement”,Computer Graphics, Vol.14, No.3, 1980

9) T.W. Calvert, J. Chapman, A. Patla：“Aspects of the Kinematic Simulation of Human Movement”,IEEE CG&A, 1982.11

10) R. Featherstone：“The Calculation of Robot Dynamics Using Articulated-Body Inertias“, The International Journal of Robotics Research, Vol.2, No.1, 1983

11) C.R. Feynman：Modelling the Appearance of Cloth, MSc thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1986

12) A. Fournier, W.T.Reeves：“A Simple Model for Ocean Waves”, Computer Graphics, Vol. 20, No.4, 1986

13) A. Fournier, J. Bloomenthal, P. Oppenheimer, W.T. Reeves, A.R. Smith：“The Modeling of Natural Phenomena”,SIGGRAPH'87 Course Notes,16, Anaheim, Calif.,1987

14) M. Girard, A.Machiejewski：“Computational Modeling for the Computer Animation of Legged Figures”,Computer Graphics, Vol.19, No.3, 1985

15) J.K. Hahn：“Realistic Animation of Rigid Bodies”, Computer Graphics, Vol.22, No.4, 1988

16) D. Haumann：“Modeling the Physical Behavior of Flexible

Objects SIGGRAPH'87 Course Notes, 17, Anaheim, Calif., 1987
17) J.M. Hollerbach：An Iterative Lagrangian Formulation of Manipulator Dynamics, MIT, AI Lab.,A.I. Memo, No.533, Revised March, 1980
18) J.M. Hollerbach：“A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation”, IEEE Transaction on Systems, Man and Cybernetics, SMC-10,11, 1980
19) P.M. Isaacs, M.F. Cohen：“Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics”, Computer Graphics, Vol.21 No.4, 1987
20) J.U. Korein, N.I. Badler：“Techniques for Generating the Goal-Directed Motion of Articulated Structures”,IEEE CG&A, Nov.,1982
21) J.Y.S. Luh, M.W. Walker, R.P.C. Paul：“On-line Computational Scheme for Mechanical Manipulators”,Journal of Dynamic Systems, Measurement, Control, 102, 1980
22) R.V. Lundin：“Motion Simulation”,Proceedings of NICOGRAPH '84, 1984
23) M. Moore, J. Wilhelms：“Collision Detection and Response for Computer Animation”,Computer Graphics, Vol.22, No.4, 1988
24) D.R. Peachey：“Modeling Waves and Surf”,Computer Graphics, Vol.20, No.4,1986
25) J.C. Platt, A.H. Barr：“Constraint Methods for Flexible Models”, Computer Graphics, Vol.22, No.4, 1988
26) W.T. Reeves：“Particle Systems—A Technique for Modeling a Class of Fuzzy Objects”,Computer Graphics, Vol.17, No.3, 1983
27) D. Terzopoulos, J. Platt, A. Barr, K. Fleischer：“Elastically Deformable Models”,Computer Graphics, Vol.21, No.4, 1987
28) D. Terzopoulos, K. Fleischer：“Modeling Inelastic Deformation：Viscoelasticity, Plasticity, Fracture”,Computer Graphics,Vol.22, No.4, 1988
29) J. Weil：“The Synthesis of Cloth Objects”,Computer Graphics, Vol.20, No.4, 1986
30) J. Wilhelms, B.A. Barsky：“Using Dynamic Analysis to Animate Articulated Bodies Such As Humans and Robots”,Proceedings of Graphics Interface 85, Montreal,Canada, 1985
31) J. Wilhelms：“Toward Automatic Motion Control”,IEEE CG&A, Vol.7, No.4, April, 1987
32) J. Wilhelms：“Using Dynamic Analysis for Animation of Articulated Bodies”,IEEE CG&A, Vol.7, No.6, June, 1987
33) J. Wilhelms：“Dynamics for Everyone”,SIGGRAPH'87 Course Notes,10, Anaheim, Calif.,1987
34) A. Witkin, K.Fleischer, A. Barr：“Energy Constraints on Parameterized Models”,Computer Graphics, Vol.21, No.4, 1987
35) A. Witkin, M. Kass：“Spacetime Constraints”,Computer Graphics, Vol.22, No.4, 1988
36) L. Yaeger, C. Upson, R. Myers：“Combinating Physical and Visual Simulation—Creation of the Planet Jupiter for the Film “2010””, Computer Graphics, Vol.20, No.4, 1986
37) 岩田，小野里：「状態モデリングへの物理制約の導入」，昭和63年度精密工学会秋期大会学術講演論文集，1988
38) 寺沢，柴本：「変形による衝突回避のアニメーション手法」，第4回NICOGRAPH論文コンテスト，1988
39) V.D. バージャー，M.G.オルソン：「力学＝新しい視点にたって」，培風館，1975
40) M.ブラディ他編：「ロボット・モーションII—動力学編」，HBJ出版局，1985

モデムの常識が変わりました!!

公衆回線で3万bpsを実現!!

マイクロコムの革新技術は、MNPクラス9の技術を使ったQX/V.32cで最大38,400bpsを実現しました。当社主催の公開実験でも、ユーザーのCADデータを弊社のファイル転送ソフト“ファースト”を使って35Kbpsを記録しています!!
まさに、ISDNもびっくりの高速性です。是非とも、テストしてみて下さい。
QX/V.32cは、専用線モデムとしても最高に高性能で安価です。

MNPは、世界の標準です。
全世界で、MNPは標準モデムとして使われています。現在、V.42として使えるのは、MNPモデムだけです。現在、米国ではクラス5のベンダーは、50社ほど有ります。最近では、大手半導体メーカーがクラス5チップの供給をアナウンスしています。又、CCITTではLAP-Mを主張していたBTも、併記が決まった直後から、クラス5を彼らのノードにサポートし始めました。いつ出て来るか分からなくて、メーカー間の互換性が保証されないし、物理スピードしか期待出来なくて高価なV.42完全準拠モデムと、現行でもV.42と互換性があり大量流通、安価及びメーカー間互換性が保証されたMNPモデムのどちらがメジャーが結論は出ています。

貴社も、間違いの無い選択MNPモデムを選択下さい。
その為の下取りサービスもあります。MNPは、絶えず進化します。
MNPを採用されれば、安心です。
最高のスピード、最高のコストパフォーマンス、最高のサービスを保証します。

ネットワークシステムの事なら、ハードもソフトもシステムもヒューコムにご相談を!!
ヒューコムは、ネットワーク・アーキテクトとしても、超一流です。

超高速RS-232Cボード新発売

この度ヒューコムでは、急速に盛り上がる高速ファイル転送等のニーズにお応えし、NEC98シリーズで115Kbpsを実現できるRS-232cボードの販売を始めます。かの有名なPDSの“SIM”に対応して設計していますので、PC98シリーズ上でIBM PC-ATでの卓越した通信ソフト、及びLANソフト等がその高速性を生かしたままで少しのパッチ当てて利用可能です。本ボードは、1ボード上に2チャンネルの232cポートを持ち、基本は2枚まで対応。
定価39,800円（サンプルソフト付きデバイス・ドライバー込み45,000円）

MNP用高速ファイル転送ソフト新発売!!

（高速でCAD等の大量のデータを転送されるユーザー及び付加価値の高い仕事に従事されている方等）このソフトを使えば、MNPクラス6以上のモデムの高速性及びMNPの卓越性を完全に生かしきれます。OEMも可。例えば、PC-98で38Kbpsが本当に生かせます。プロトコルによるオーバーヘッドが殆ど有りません。計時機能も付いています。
上記、超高速RS-232Cボード対応バージョンも有ります。
ユニックス用も有ります。お問い合わせ下さい。

ただいま、代理店募集中!!

●米マイクロコム社総代理店
株式会社ヒューコム
■技術、営業スタッフ募集中！ あなたも、先端ビジネスにチャレンジしませんか？

TEL.(03)220-0535(代) FAX.(03)220-4975
（本　社）〒167 東京都杉並区荻窪5-30-12 グロリアビル
（西日本支社）〒604 京都市中京区麩屋町通三条下ル白壁町444 ☎075(256)0261 FAX.075(256)0262
●東北地区代理店 (株)オーエックスエンタープライズ情報機器部 TEL.(022)262-5540 〒980 仙台市一番町4-6-1

ベンチャー企業がカスタムICを使ってGWSを作る時代だ

ジョン・ウィリアム・ポダスカ会長
（米国ステラ・コンピュータ）

グラフィック・ワークステーションは，CAD／CAM，CAE，コンピュータ・グラフィックスをインタラクティブに扱うためのコンピュータとして確立され，市場でも急激に伸長している。その中でも，特にハイエンドのグラフィックス環境を構築するためのコンピュータであるグラフィック・スーパーワークステーションを市場に投入している日米のベンチャー企業，米国ステラ・コンピュータのジョン・ウィリアム・ポダスカ会長と日本電算機の石井孝利社長に，CGのキー・テクノロジと今後の展望をテーマにして語っていただいた。石井社長はグラフィック・ワークステーション専業メーカーである日本電算機の若きトップとして世界に目を向けている。ポダスカ会長はスーパー・ミニコンのプライム コンピュータ，エンジニアリング・ワークステーション・ベンダーであるアポロコンピュータの創設者でもある。現時点ではそれほど大きくないこの市場をどのように広げていくのか，技術を見すえた起業家である2人が熱っぽく語り合った。

石井孝利社良（日本電算機）

▶ カスタムICの技術によってCGが成熟しつつある

PIXEL　本日はお忙しいところをありがとうございます。お2人とも日米におけるハイエンドなグラフィック・ワークステーションのリーダーとしてご活躍中ですが，まず今後のワークステーションをベースにしたコンピュータ・グラフィックスの市場動向についてうかがいたいと思います。特に，ハイエンドなワークステーションの市場は非常に魅力的であり，その使い方や仕様，価格などについてはこれから固まっていくものと思われますが。

ジョン・ウィリアム・ポダスカ（以下ポダスカと略す）　コンピュータ・グラフィックスはアメリカではようやく成熟してきた産業ですね。以前から製品はありましたが，これまではやや高価でした。ここ2年で製品も低価格化し，大変良質になってきたと思います。

カスタムICやASICなどの技術が日本で熟成されてきて，特に生産技術に優れた日本のメーカーが力をつけて高度な製品を低価格かつ良質に作れるようになったことが，コンピュータ・グラフィックスの市場が広がってきた大きな理由と思われます。石井さんの会社でも同様だと思いますが，われわれのアーキテクチャの中でも多用されているようにカスタムIC技術や大容量フレーム・バッファによりコンピュータ・グラフィックスの製品が低価格となりました。日本の市場については門外漢ですが，当社の日本法人や販売代理店もこの市場におけるビジネスで大きな成功を収められると期待しています。

石井孝利（以下石井と略す）　これまではグラフィックスはグラフィックス，コンピュータはコンピュータというように2つの世界は別々に展開されてきたと思います。そこへネットワークの発達によりワークステーション化のブームが巻き起こりました。ところが，現在のワークステーションのユーザーはさらに大きなCPUパワーをもった高速なワークステーションと，その出力をグラフィック化したいという2つの新しい要求を感じ始めているわけです。今後3年間，ユーザーはCPUの高速化とレンダリングの多様化の必要性を感じ，よりパワフルなグラフィック・スーパーコンピュータを求めていくでしょう。

▶ 新技術の出現がベンチャーの成功を助ける

PIXEL　一般的に申し上げて，ワークステーション分野だけでなく，コンピュータ・グラフィックス分野全体でいわゆるベンチャー企業の進出が著しいわけです。ベンチャー企業が多く進出するには，これまでの既成企業が進出しない，またはできにくいことについてそれなりの理由があるはずですが，これま

でベンチャー企業を成功させてきたポダスカさんはその大きな要因を何だとお考えですか?

ポダスカ　これまでの既成の技術を大きく改善する余地があったり，その新しい市場の伸びが確認できれば（ベンチャー企業が進出して）ビジネス・チャンスをつかむ機会が十分にあるといえるでしょう。ベンチャー企業をスタートさせるタイミングとしては，ステラ・コンピュータの場合はカスタムIC技術の発達が要素の一つとなっています。GS1000ではカスタムICを61個，データロジックを250万個も使用していますが，このようなことは3年前は不可能でした。ステラ・コンピュータはこの技術をベースとしてシステム・デザインを可能としました。ここでも日本の技術が可能性を高めたと考えています。比較的資本力の小さなベンチャー企業がカスタムICを手軽に手に入れられるということは，ベンチャー企業にとって大きな福音です。

このような新技術が出現するということは，一般的にみてベンチャー企業が成功する可能性が残されているといえるでしょう。1980年にはアポロコンピュータとサン・マイクロシステムズが，同じようにシングルボード・マイクロプロセッサによって飛躍的な成長を遂げました。この他のベンチャー企業をスタートさせるタイミングについても同様のことが当てはまるでしょう。

石井　まさにおっしゃる通りで，日本電算機でも60個以上のカスタムLSIを活用しています。コンピュータ・グラフィックスの最前線ではレンダリングを含めた表示スピードの速さや，レンダリング・バリエーションの豊富さが要求されてきています。グロー・シェーディング，フォン・シェーディング，アンチ・エイリアシングが利用されており，私もこれらの新しい要求に対してはカスタムIC，ASICのような最新の技術が解決におおいに役立っていると考えています。ということは，カスタムICやASICといった技術が今後の新しい技術およびベンチャー企業の創造に及ぼす影響が非常に大きいということですか。

ポダスカ　その通りです。

ベンチャー企業は新しい市場を自らの努力で広く大きくしていく必要がある

石井　ミニ・スーパーコンピュータ，グラフィック・スーパーコンピュータについてですが，日本のベンダーはアメリカとは異なり，この分野に対する取組みが消極的です。こうしたニュー・ニッチ・マーケット（新しい隙間市場）はどのように開拓していくべきなのでしょうか。

ポダスカ　日本のニッチ・マーケットが小さいということを心配する必要はありません。確かにアメリカは市場全体が大きいため，ニッチ・マーケットのサイズもそれだけ大きいといえます。しかしベンチャー企業の戦略は（現在ある市場だけを対象にするのではなく）ニッチ・マーケットを作っていき，大きく育てていくということです。そのニッチ・マーケットにくさびを打ち込み，深く広くしていくものです。アポロコンピュータのワークステーション戦略を例にとりますと，最初は「1人1台の時代」をうたって1台3万ドルのマシンを開発しました。次に，これにカラー・グラフィックスを加え，機能・性能の向上を図って1台6万ドルの製品を出しました。その後，価格をまた1台3万ドルまで下げたのです。このようにして市場をだんだんと大きくしていったのです。

ステラ・コンピュータも同じ方法をとると思います。第1段階としてはハイエンド・グラフィック・スーパーコンピュータで基礎をつくり，次々に市場を拡大していくことでその後のハードウエア，ソフトウエアを出していくことができるのだと思います。製品の購買力が伸び，ハイエンド・グラフィック・ワークステーションができて，これがまた低価格になるというサイクルで市場を広げていけるはずです。

グラフィック・スーパーの各メーカーは企業ごとに独自の市場を作り出そうとしている

PIXEL　ワークステーションの定義も当初ワークステーションが提案されたときから，これは当然のことですが，その仕様などは大きく変わってきています。現在はワークステーション，エンジニアリング・ワークステーション，パーソナル・ワークステーション，グラフィック・ワークステーション，スーパー・ワークステーション，グラフィック・スーパーワークステーションなどといろいろな呼び名があります。ユーザーもあまりに多くの名称があることでやや困惑しているほどです。それだけ市場が大きくなりつつあるということでしょうが，今後のワークステーションがどのような方向にいくか，大きな期待がありますね。

ポダスカ　ワークステーション全体はかなり広義なものになってきました。ワークステーション市場へ参入する企業が増えたのと，技術自体が変化したことがその理由です。ステラ・コンピュータの場合も，今年（1988年）の新製品はその第1歩にすぎません。2倍の性能をもつ製品を次の年には発表し，再来年にはさらにその2倍にできるはずです。競合する企業はアメリカ，日本，ヨーロッパで増えてくるでしょう。そういった状況の中で，われわれも含めたすべてのメーカーは各メーカーごとに独自の市場を見いだそうとしています。例えば，シリコングラフィックスはIRIS POWERというハイエンド・マシンを投入する一方で，非常に低価格なパーソナル・ワークステーションであるPersonal IRISを出してきました。アーデント・コンピュータのTITANシリーズはグラフィックスよりも演算能力の方に焦点をしぼっているようです。そして，ステラ・コンピュータは演算能力とグラフィックス能力のバランスをうまくとっていこうと考えています。

これはどこが正しくどこが間違っているというのではなく，みんながそれぞれ正しいのだと思います。これから新しい技術が開発されていけば，さらに市場の分化が進み，大きくなっていくでしょう。飛行機を例にとると，50年前に飛行機が作られ

た当時，多くの企業が参入しましたが，マクダネル・ダグラスは軍事用に，ボーイングは旅客用にと異なる道を歩みました。旅客会社にしてもパンナム，TWA は国際線へ，ノースイースト，イースタンは国内線へと分化，集中していきました。日本電気，日立製作所，富士通，東芝など半導体の世界でも同じようなことがいえるのではないでしょうか。富士通は ASIC に強く，東芝はメモリに強いといった具合にです。

石井　新しいマーケットの多様化と特徴をもった各社による新しいマーケット・セグメンテーション，これが今後の傾向でしょう。日本市場は多くの競合会社が次々に参入し，ベンダーとしては大変厳しい状況になると思われます。各社の棲み分けは今後の日本市場でも重要なことになると思います。

◐ 新会社を設立しても前の会社と依然として良好な関係にあるのがアメリカのビジネスのルールである

PIXEL　ポダスカさんはすでに 3 つの会社を設立して 2 つの会社をスピンアウトしているわけですが，新会社を設立するためにはなんらかのアイデアと，それを実現するためのタイミングがあったと思います。どのようにすればそのようなアイデアやタイミングがつかめるのでしょうか。

ポダスカ　一般的にいって，新しいアイデアはこれまでの企業の中で熟成されるものです。プライムコンピュータではさまざまなコンピュータのアイデア，マーケットのアイデアが考え出されました。アポロコンピュータにおいても同様です。新しいアイデアが新しいマーケットで生まれるのは稀なことなのです。

プライムコンピュータをスピンアウトしアポロコンピュータを設立した 1980 年には，ワークステーションは極小市場でした。このときには，プライムコンピュータはワークステーションの開発には全く関心がなかったのです。プライムコンピュータはすでにスーパー・ミニコンを中心とする市場をもっており，タイムシェアリング・マシン，ビジネス・データプロセッサの開発をしていました。その中の小さなグループがワークステーションの開発へと手を伸ばしていったのですが，成功するかどうかはまだ疑わしい状況でした。そのため私がスピンアウトして小さな会社を作り，ワークステーション市場に参入したのです。その一方で，プライムコンピュータは従来の路線の活動を続けています。しかしながら，現在プライムコンピュータと私との関係は大変良好で，敵対する要素は全くありません。ビジネスにおいても個人的にも非常に親密です。

ステラ・コンピュータとしてはアポロコンピュータ，プライムコンピュータと現在でも良好な関係を保っています。ちょうど私が日本へ来る前にもアポロコンピュータのトム・バンダスライス社長と会食しました。それぞれがお互いの道を歩み，それぞれの領分で協力をし合っており，双方に問題はありません。アポロコンピュータの株式も多少は持っていますが，決定権があるほどの数ではありませんし，プライムコンピュータやアポロコンピュータをコントロールするつもりはありませんからね。企業機密やアイデアを盗んで新会社を設立するなどというのは全く論外ですが，例えばステラ・コンピュータから独立してパーソナルコンピュータの新しい会社，例えばアップル・コンピュータのようなベンチャーを作りたいという社員がいるなら，握手を交わして互いの幸運を祈るでしょう。

石井　それは日本とは相当に違った状況ですね。やはりアメリカがいかに自由というものを大切にしているか，いかに人間というものを大切にしているかが今のお話からもわかりますね。ステラ・コンピュータには，アポロコンピュータからポダスカさんと一緒に移ってきたスタッフはいらっしゃらないのですか。

ポダスカ　何人かはいます。協定によってアポロコンピュータなどの社員は 3 年間は雇用しないことになっています。実に公平なシステムです。ただ，ボストン地区にはレベルの高い人間が多いので雇用に関する問題はありません。

石井　このあたりは日本と違って，アメリカではルールが確立していてベンチャー企業を創設する人間とその周辺との関係が非常にうまくいっていると思われます。こうしたことは日本のベンチャー企業を考える場合に大変参考になるルールだと思います。もう一つの問題は雇用についてですが，日本でも優秀な人材がベンチャー企業に活力を提供できるようになって初めて本格的なベンチャー企業の時代を迎えることになりますね。残念ながら，現在ではそうなっていません。

◐ ステラは日本の市場に大きな期待をかけている

PIXEL　最後に，日本におけるコンピュータ・グラフィックスの市場性についてはどのような印象をおもちですか。

ポダスカ　日本の市場性については満足しています。日本はアジア，太平洋，韓国，台湾，香港，シンガポール，中国，オーストラリア，フィリピンの中心となっており，当社の 1/3 の業務を日本には期待しています。売上げでは米国が半分を少し切るくらいで，米国以外の市場に半分以上を期待しており，そのほとんどは日本です。素晴らしいビジネスになると思います。ここまで比重が高いというのはもちろんドル安のおかげもあるのですが。ドル安は私がコントロールしているわけではありませんし，レーガン大統領やブッシュ次期大統領から日本でビジネスするようにいわれたわけでもありません。大統領がブッシュ氏に決まったおかげでやりやすいというのはありますけれど。

日本とアメリカとはいろいろな面で異なる点がありますが，アメリカ人は忍耐，高品質，協調性，生産技術，在庫管理などを日本から学んでいます。一方，日本人はアメリカから規制を受けず，正しいことを全うする方法を学ぶべきだと思います。このように互いの良いところを学んで新しい両国の間柄を構築していくべきだと思います。

PIXEL　本日はお 2 人ともお忙しい中をありがとうございました。

new products

3次元アニメーション統合化ソフト

リンクス

リンクスは，3次元コンピュータ・アニメーション・システムPERSONAL LINKS NEWS版の統合ソフトウエア「ANNIE」の販売を開始した。

PERSONAL LINKSは，フルカラー・フレーム・バッファFMX-32，モデリング・ソフトウエアPOLYGON EDITOR，レイ・トレーシング・ソフトウエアTRACY，レンダリング・ソフトウエアPOLLY，ペイント・ソフトウエアPICTORISなどから構成されている3次元アニメーション・システムであり，ANNIEは各ソフトウエアの管理，呼出し，実行などを行う。

また，形状，質感，データ，動画などに対するCGデータごとのビジュアル・エディタ，グラフ表示によるアニメーション・データの編集，ワイヤー・フレームによる高速プレイバック，オンメモリ・シェーディング・アニメーションによる低解像度ディスプレイでの高速プレイバックなどの機能がある。

価格は300万円となっている。

問合せ先：株式会社リンクス　事業部
〒140 東京都品川区東品川3-13-6
☎03(450)8163

カラーデザイン・ソフトウエア

精工舎

精工舎は，カラーデザイン・ソフト「Techno-CANVAS」の販売を開始した。

同製品は，Techno-PROシリーズ，FA386，FA286 IIなどで動作するペイント・ソフトであり，2色3方向のグラデーション，モノクロ8種・カラー8種の計16種類のスクリーントーン，指定矩形の色の瞬時変換，データに対するマスク機能などの各種機能をもっている。また，スキャナから入力した画像，ワードプロセッサ，パース図作成システムなどとデータ交換を行うこともできる。

価格は10万円となっている。

同社では，建築，アパレル，パッケージ，家具，車，美容，広告，一般文書などへの需要を見込んでいる。

問合せ先：株式会社精工舎　情報機器事業本部　〒130 東京都墨田区大平4-1-1
☎03(621)4011

リアルタイム描画が可能なパース図作成ソフトウエア

セイコー電子工業

セイコー電子工業は，高速3次元グラフィック・ディスプレイの機能を生かし，本格的景観シミュレーションをリアルタイムで行うことができるパース図作成ソフトウエア「SG-PERS」の販売を開始した。

従来はリアルな表現とリアルタイム性の両立は困難であったが，SG-PERSは3次元グラフィック・ディスプレイGR-4416の機能を利用することにより実現した。GR-4416のローカルな機能として，3次元シェーディング機能，ライティング機能，ビューイング機能などをもっているため，グラフィック表示の際にかかるホスト・コンピュータの負担が少なくなり，描画速度が向上し，リアルタイムで色付け，光源，視点位置などの変更を行い，確認することができる。

また，外壁の色や材質感，季節による太陽光の違いによる変化，建物の移動，回転，拡大，縮小などの各種機能の他，レイ・トレーシング・ソフトウエアSG-RAYとのインタフェースをもっているため，レイ・トレーシングでのリアルな表現も可能である。操作は，日本語によるメニュー選択方式を採用した。

価格は380万円となっている。

問合せ先：セイコー電子工業株式会社
情報関連事業部電子機器2部
〒136 東京都江東区亀戸6-31-1
☎03(684)5722

リアルタイム画像処理システム

伊藤忠テクノサイエンス

伊藤忠テクノサイエンスは，米イメージング・テクノロジー社のイメージプロセッサ「Series151」と，米アプライド・メモリ・テクノロジー社の画像処理用ディスク「モデル8300 リアルタイム・ディスクシステム」の販売を開始した。

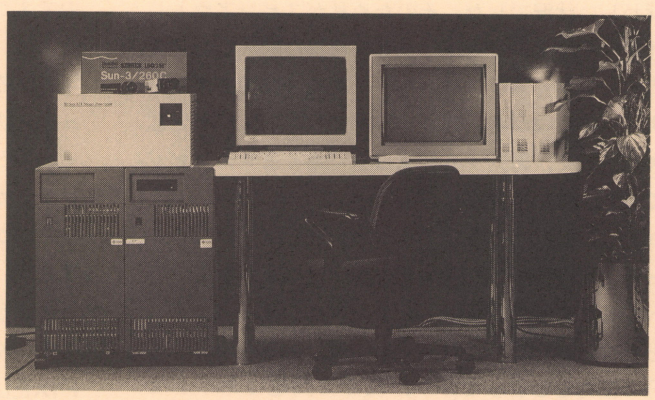
Series151は，SunなどのVMEバス・ベースのワークステーションやPC/AT，PS/2などと接続できるイメージ・プロセッサである。また，各種の画像処理用モジュールと組み合わせて，R&D，マシンビジョン，医療などの画像処理アプリケーションに対応できる。画像処理用モジュールは，ユーザーのシステムごとにボード単位での提供も可能であるため，開発システムから目的システムまでの一貫したアーキテクチャの構築がローコストで行える。

グラフィックス解像度512×512×8ビットの画像に対し，四則演算，論理演算，コンボリューション，ヒストグラム抽出，特徴抽出，メディアンフィルタ，モフォロジー，パイプラインなどの処理を1/30秒で実行する。また，オプションで20MFLOPSのアレイプロセッサ・ボードも用意されており，FFT，ワープなどの処理の高速化が図れる。付属ソフトウエアとしてSun用のインタプリタを含むライブラリが提供される。

モデル8300は，512×512×8ビットの画像データを30画面/秒の読み書きができ，記録した画像に対して再生，逆再生，フリーズなどの処理を行える。記憶容量は，340Mバイト(動画33秒間)，680Mバイト(動画67秒間)，1.02Gバイト(動画100秒間)の3タイプが用意されている。

両製品は，同時に使用することができるため，同社では輸入販売を行っているSunシリーズなどと組み合わせて医療用，FA用などの画像処理システムやフルカラーのアニメーション・システムへの需要を見込んでいる。

価格は，Series151が基本システムで236万円から，画像処理用モジュールが41万円から，モデル8300が340Mバイト・ディスクタイプで626万円，同680M



バイトが812万円，同1.02Gバイトが998万円となっている。

問合せ先：伊藤忠テクノサイエンス株式会社　半導体システム事業部
〒154 東京都世田谷区駒沢1-16-7
☎03(419)9190

パソコン用 フレキシブルCAMシステム

アンドール

アンドールは，NC加工対応のフレキシブルCAMシステム「CAM CORE」の販売を開始した。

同製品は，各種金型，機械部品加工，レーザー加工，土木加工などの各種業務とマシニング・センター，NCフライス，ワイヤー・カット，レーザー，土木機械などの各種の加工機に対応したCAMソフトウエアであり，ユーザーのカスタム化が容易に行える。また，IGES，DXFなどの汎用画像フォーマットに対応していると同時に，同製品自体の図形データフォーマットが公開されているため，ユーザーがカスタム化してきたCADシステムのデータがそのままCAMシステムの入力として使える。

加工機能は，周径加工，形状加工，島残し加工，等高線加工，路上加工，穴加工，3次元加工など各種がサポートされている。また，BASICライクなインタプリタ言語により，加工に必要な道具，材料加工条件などはすべてユーザーに合わせて設定できる。

価格は，基本部90万円，IGESインタフェースなどの各種オプションが30万円からとなっている。PC-98XA/XL/XL²で動作する。

問合せ先：アンドール株式会社　CAEシステム部CAM開発グループ
〒650 兵庫県神戸市中央区中町通2-3-2
☎078(351)5241

フルカラー写真 データベースシステム

MISAWA・VAN

MISAWA・VANは，カラー写真をファイリング，検索，表示できるカラー写真データベースシステムの販売を開始した。

同製品は，カラー写真，文字，グラフィックスを同時に管理できるデータベースシステムであり，カラー写真は1,677万色の同時表示が可能である。データの検索，表示などの指定はすべてCRT上に装着されたタッチパネルによって行える。

入力信号は，ビデオカメラ，スチールカメラ，VTR，ビデオディスクなどのNTSC信号とパーソナルコンピュータのRGB信号に対応しており，カラー画面は専用のカラー写真プロセッサによりフルカラーで表示される。また，画像は1/16，1/8，1/4の3段階の圧縮を行うことができる。

ソフトウエアは，カード型データベースの「フォトクリッパー」と，階層構造型データベースの「フォトファイル」が準備されている。

専用の高速モデムを使用してオンラインでの利用もできる。データベースには固定ディスク，光ディスクのいずれも使用でき，20Mバイトのディスクで約400枚の写真のデータベースが構築できる。

システムは，カラー写真プロセッサ，フォトファイルまたはフォトクリッパー，それにタッチパネルから構成されており，PC-9801シリーズのMS-DOS上で作動する。

価格は，カラー写真プロセッサが139万3,000円，フォトファイルが38万円，フォトクリッパーが58万円，タッチパネルが31万8,700円，オプションの高速モデムがインタフェース・ボード込みで35万5,500円となっている。

同社では，ショールームやロビーなどでの商品紹介，結婚式場やホテルでの案内システム，美術館や博物館での展示品ガイドシステム，企業の資料室での写真アルバムシステムなどへの需要を見込んでいる。

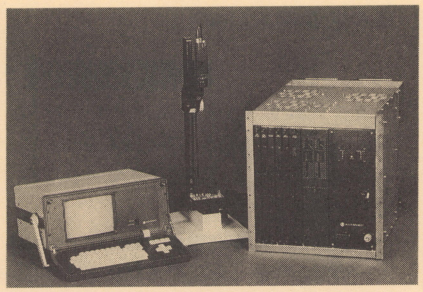
問合せ先：MISAWA・VAN株式会社　コンピュータ技術部
〒160 東京都新宿区新宿3-24-1
☎03(352)4511

汎用画像認識装置

エヴィック

エヴィックは，米アレン・ブラッドレー社製汎用画像認識装置「エキスパートPVS2805」の販売を開始した。

同製品は通常の画像処理装置と異なり，画像認識形態学(モフォロジー)を初めてハードウエアに採用した，高速並列処理が可能な汎用性・拡張性を備えた画像認識装置であり，半導体ウエハーの個体識別，機械部品の加工・組立検査，部



品の寸法検査，プリント基板外観検査，組立ロボット・ガイダンスなど欧米の工場自動化ですでに実績がある。

入出力は，ビデオカメラ，パラレルインタフェース，RS-232Cなどに対応しており，コ・プロセッサ処理で画像の取込み処理，外部装置との通信が同時に行える。グレースケールおよび二値化画像を含めた入力画像に対して強調処理，フィルタリング処理，画像再構成，構造解析などの処理が行える。また，業務に応じてプロセッサ/メモリ基板を追加できる。

価格は，製品構成により300万～800万円となっている。

問合せ先：エヴィック株式会社　マーケティング部
〒153 東京都目黒区青葉台4-4-5
☎03(467)7321

PostScript対応のフォント

アイコン

アイコンは，グループ会社であるユーミーズ社開発のPostScriptフォントの販売を開始した。

同製品は，米国で主流となっているアドビ社のページ記述言語PostScriptによって記述されたベジエ曲線のアウトライン・フォントであり，PostScript対応のプリンタであれば，6ポイント(約2mm四方)から約300mm四方の大きさまでの文字を出力できる。

今回発表されたフォントは，明朝体，ゴシック体，江戸文字の3種類であるが，同社では，全部で12種類・約60書体のフォントを提供する予定である。

価格は，字母として切り売りする場合で2,000万～5,000万円，ソフトハウスなどへのロイヤリティ販売の場合，基本料金の他，販売量に応じたロイヤリティを取る仕組み。

問合せ先：株式会社アイコン
〒185 東京都国分寺市本町3-10-22
☎0423(26)0085

new products

日本語版 MacDraw II

システムソフト

システムソフトは，米クラリス社のMacintosh用作図ツールMacDraw IIの日本語版である「マックドローII」の販売を開始した。

同製品は，約500層までのマルチレイヤー機能，作成した図形を部品として登録するライブラリ機能，設定縮尺率や設定単位に合わせての自動採寸機能，線のパターンや太さのカスタマイズ機能などの多くの機能をもつ作図ツールであり，プレゼンテーション用資料作成，DTP，建築・機械設計などの簡易CADなどに米国で使用されている。

日本語化では，Macintoshの漢字フロント・プロセッサである漢字Talk2.0への完全対応と，大部分の表示の日本語化を行った他，定規の作図単位をインチ中心からセンチメートル中心の設定にしたり，ペン幅の作図単位をポイントからミリ単位に変更するなど日本での使用環境に合わせている。また，日本語の注釈を記入して，その部分にアンダーラインを引くことができる。デフォルトのフォントはOsaka12ポイントとなっている。

出力は，Laser WriterなどのMacintosh用プリンタの他，各種プリンタ/プロッタに対応している。動作には，Macintosh II/SE/Plusのいずれかの機種と漢字Talk2.0と空きメモリ1Mバイトが必要である。

価格は7万円となっている。

問合せ先：株式会社システムソフト
〒810 福岡市中央区天神5-7-2
☎092(714)6236

ロボット・シミュレーション用システム

住商エレクトロニクス

住商エレクトロニクスは，米国デネブ・ロボティクス社製でファモティク社が輸入しているロボット・シミュレーション用システム「IGRIP」の販売を開始した。

同製品は，工場のFA化に伴い，産業用ロボットが急速に増えると同時に，ロボット管理や安全動作のための技術が複雑になってきたことに対応したもので，ロボットのティーチングを行う。

対話形式でのモデリング機能でロボットの形状を入力したり，IGESによる入出力機能を利用し上位CADシステムからデータを入力して形状の作成ができる。新規にユーザーが定義したロボットに対して運動特性(キネマティックス)を与え，その運動特性に対して回転率，移動系，駆動速度などの属性を与えることができる。また，運動特性のチェックのためのJOG機能もある。

シミュレーションは，複数のロボットの協調動作が可能であり，正確なサイクルタイムの計算ができる。Pascalライクなシミュレーションのための制御言語IGRIP/GSLが準備されており，ユーザーは独自のシミュレーション環境を構築することができる。同社で販売している米シリコングラフィックス社製のGWS IRISシリーズをハードウエアとして採用したことにより，従来のシステムに比べてよりリアルタイムにシミュレーションが行える。

価格はハードウエアを含めて約2,000万円となっている。

問合せ先:住商エレクトロニクス株式会社
CAD営業部
〒101 東京都千代田区神田錦町3-1-1
☎03(292)2411

データ解析用グラフィック・ソフトウエア

ニチメン

ニチメンは，東洋情報システム社とその販売会社であるティアイエスシステムエイジと共同で，米プレシジョン・ビジュアル社のグラフィック・ライブラリ「GFX-4000」と，データ解析ソフトウエア「PV～WAVE」の販売を開始した。

GFX-4000は，3次元グラフィックス標準であるPHIGSの機能を取り入れるとともに，独自の機能をもったグラフィック・ライブラリである。PHIGSと同様，ウィンドごとに複数のビューが設定できるが，プロセスごとに複数のウィンドを作ったり，PHIGSでいうワークステーションごとに複数のウィンドをオープンすることができる。また，最大のパフォーマンスを得るためにハードウエアの機能を多用しているが，ハードウエアに依存しないためのシミュレーション・ライブラリも用意されていると同時に，ライブラリはすべてC言語で書かれているため移植が可能である。ライブラリのアプリケーション・プログラム・インタフェースはCとFORTRANをもっている。

基本プリミティブはPHIGSで定義してあるものと，三角形，円，平行四辺形などの独自のものをもっている。入力機能はPHIGSの各機能とウィンドをサポートしている。また，ラスター・オペレーション，隠線・隠面処理もサポートしている。同社では，各種データ解析アプリケーション用としての需要を見込んでいる。

PV～WAVEは，ワークステーション上で対話処理方式でデータの解析から表示までを行うソフトウエアであり，実験結果や測定データなどのデータを読み込み，さまざまな処理や変換を行ったうえでグラフィックス表示を行う。イメージデータの表示，ヒストグラムによる分析，組込み関数およびユーザー定義関数などの各種の処理が簡単に行える。同社では，物理学，天文学，医学，航空宇宙学，エンジニアリングなどの分野での需要を見込んでいる。

両製品とも現在はVAXstationシリーズ用のみ販売されており，今後Sunなどのワークステーション用に移植する予定である。

価格は，GFX-4000が100万円，PV～WAVEが150万円からとなっている。

問合せ先：ニチメン株式会社　電子情報機器部情報機器課
〒103 東京都中央区日本橋3-11-1
☎03(277)5851

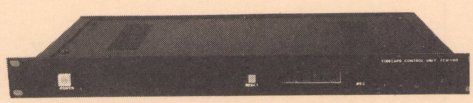
低価格CG画像のコマ撮りシステム

シービー通商

シービー通商は，CGの画像などをコマ撮りできるVTRを含むシステムの販売を開始した。

従来，3/4インチや1インチなどでコマ撮りできるシステムはあったが，同製品はVHS規格のVTRを制御することによりシステムの低価格化を実現した。画質は1インチや3/4インチに比べて劣るが，ローコストでCGアニメーションを作成する場合などに利用できる。

また，コンピュータのディスクなどにファイルされているCG画像を呼び出しながらVTRにコマ撮りするため，人が付いている必要はない。1コマを1フレームまたは4フレームのいずれかに設定できる。



new products

システムは，コマ撮り専用VHS VTRとモニター用14型テレビ，プレゼンテーション用ポータブルVTRと6型ポータブルテレビ，VTRコントローラから構成されている。

価格は120万円となっている。

問合せ先：有限会社シービー通商
〒173 東京都板橋区中板橋15-5
☎03(962)7341

32ビットDTPシステム

JDL

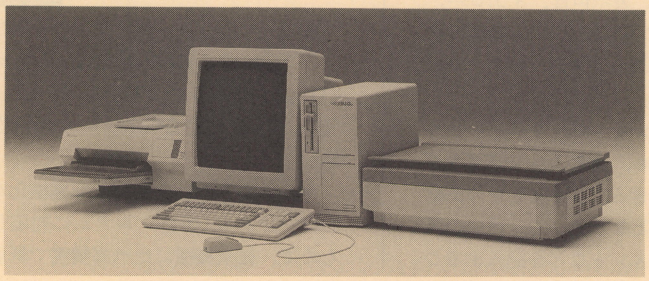
日本デジタル研究所は，文作くんシリーズの新機種であるDTPシステム「文作-30DTP」の販売を開始した。

システムは，CPUにMC68020，数値演算プロセッサにMC68881を採用したホスト・コンピュータに，主記憶3Mバイト，19インチCRTディスプレイ，レーザー・プリンタとソフトウエアから構成されている。ディスプレイは1,280×1,280ドットの高解像度で，16ドット文字で79行×60文字の表示ができるため，A4判の用紙の縦，横の表示ができる。レーザー・プリンタは400dpiの解像度で，A4判の用紙を8枚/分出力できる。また，オプションで写真やイラストが読み込める400dpiの解像度のイメージリーダーの接続が可能である。

文書作成は，システム文作パブリッシングを機能強化したものを搭載し，段組，ページ内の縦書き・横書き混在，回転文字など従来困難だった操作もできる。フォントは，ドット・フォント，ベクトル・フォントのどちらにも対応している。ベクトル・フォントでは4〜70ポイントの表現が，各文字に対して斜体，回転体などの表現ができる。また，LANシステムの中に組み込むこともできる。

価格は225万円となっている。

問合せ先：株式会社日本デジタル研究所
経営企画室
〒163 東京都新宿区西新宿2-1-1
☎03(348)6751



地籍情報管理システム

桜井

桜井は，パーソナルコンピュータをベースとした地籍情報管理システム.「LANDMAN」の販売を開始した。

このシステムは，国土庁の標準フォーマットによる国土調査成果のデータ作成を行い，作成したデータベースをもとに測量設計，計算，製図などの処理を行う。データは40Mバイトの固定ディスクに格納され，検索，修正，追加なども簡単に行うことができ，事業計画から，計算，集計業務，認証，登記業務までの作業を一貫してできる。

同社では，市町村の土地行政業務の合理化，住民サービス，登記業務，固定資産税務などへの需要を見込んでいる。

システムはホスト・コンピュータにIBMのパーソナルシステム/55モデル5550を採用，プリンタ，ストリーマ，タブレット，X-Yプロッタ，それに独自に作成したソフトウエアを組み込んでいる。

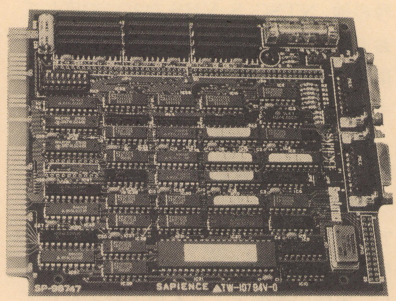
価格は518万円である。

問合せ先：桜井株式会社
〒110 東京都台東区東上野1-11-12
☎03(835)4311

CAD/DTP用インタフェース・ボード

サピエンス

サピエンスは，通常のPC-9801で高解像度(ハイリゾレーション)モード用ソフトウエアを動かすCAD/DTP用インタフェース・ボード「SuperLattice」の販売を開始した。



同製品は，CAD，DTPなど多くのソフトウエアでPC-98XA/XL/XL²のみがサポートしている高解像度のモード用に作られたものであるため，通常のPC-9801で使えるようにした。このボードを拡張スロットに差し込んでマルチシンクのディスプレイを用いると，高解像度対応ソフトウエアが使用できる。また，オプションでSP98745を追加すれば，デフォルトの8色表示から1,670万色中256色の表示ができる。また，ピクセル単位の書込み，プレーン単位の書込みのいずれも可能で，プレーン単位の書込みの場合8または16ピクセルの同時書込みができる。

価格は15万8,000円である。

問合せ先：株式会社サピエンス
〒169 東京都新宿区高田馬場4-9-11
☎03(367)0734

2次元データから3次元モデルを作るシステム

ソニー・テクトロニクス

ソニー・テクトロニクスは，入力した2次元形状から3次元アニメーションを作成することができるシステム「3Dグラフィック・デザイン・クリエータ」の販売を開始した。

同製品は，2次元CADソフトTekniCADで作成したデータ，マウス，タブレットを用いてリアルタイムで作成した図形などの2次元形状を入力し，その形状をもとにして任意の厚み付け，回転，曲線のパイプ化などの処理を行ったうえで3次元のモデルを作成するソフトウエアであり，EWSの4337型の上で使用できる。作成した3次元モデルは4337型のグラフィックス機能を用いて各オブジェクトごとの回転，移動，拡大，縮小などの処理を行うこともできる。

また，レンダリング・ソフトWavefrontとのインタフェースも可能な他，VTRコントロール用のアニメ・コントロール・ユニットを使用することによりビデオ・アニメーションの作成もできる。

価格はソフトウエアのみで95万円となっている。

問合せ先：ソニー・テクトロニクス株式会社　情報機器部
〒141 東京都品川区北品川5-9-31
☎03(779)7602

new products

EWSで動く プリント基板設計システム

日本IBM

日本アイ・ビー・エムは，図研社開発のプリント基板の設計から製造までを一貫して支援する統合化ソフトウエアを，EWSに移植した「IBMプリント基板設計システムCR-3000」の販売を開始した。

同製品は，すでに業務実績があるプリント基板の設計準備，配置設計，配線設計，アートワーク設計，設計検証，帳表作成，製造データ作成といった一貫設計・製造ができるソフトウエアであり，EWSであるIBM6100プロフェッショナル・ワークステーションで稼働する。また，EWSで稼働するためマルチウィンド処理ができ，アナログ，デジタル，アナログ/デジタル混在の各プリント基板を簡単に設計し，製造することができる。

システムは，基本となる対話型のアナログ基板設計基本モジュール，アナログとデジタル混在のHIC(ハイブリッドIC)基板設計用対話型モジュール，データをプロッタに出力するためのプロッタ・サポート・モジュール，設計データからNC機器用データを作成するCAMモジュール，プリント基板を設計するフォト・プロッタのデータを取り込むためのNCサポート・モジュールから構成されている。

価格は，アナログ基板設計基本モジュールが890万円，HIC基板設計用モジュールが390万円，プロッタ・サポート・モジュールが80万円，CAMモジュールが230万円，NCサポート・モジュールが130万円となっている。

問合せ先：日本アイ・ビー・エム株式会社
〒106 東京都港区六本木3-2-12
☎03(586)1111

久保田鉄工と共同で J-3100版CADKEYを販売

日本オリベッティ

日本オリベッティは久保田鉄工と共同で，米キャドキー社の3次元パーソナルCAD「CADKEY」の販売を開始した。

同製品は，2次元の製図，3次元の設計システムを完全に結合したパーソナル3次元CADシステムであり，3次元データは独自のデータベースに格納されている。標準でワイヤー・フレームでの，オプションでソリッド・モデリングでの表現が可能となっている。また，データベース構造は公開されているので，インタフェース言語CADLやCK-IGESトランスレータを用いて，ユーザーのカスタマイズや外部プログラムとのリンクが可能な他，マクロ定義などもサポートされている。

CADKEYは，米国を中心に世界で約45,000セットの売上げをもつ3次元パーソナルCADソフトウエアであり，久保田鉄工が日本国内で総販売代理権をもっているが，今回，日本オリベッティ東芝のパーソナルコンピュータJ-3100用CADKEYの販売を開始した。

価格は，J-3100SGTとのシステムで190万～370万円，ソフトウエアのみは英語版が75万～107万円，日本語版が110万～160万円となっている。

問合せ先：日本オリベッティ株式会社
〒152 東京都目黒区碑文谷1-26-17
☎03(714)1211

フルカラー・オートスキャン型 高解像度・低価格ハードコピー

三菱電機

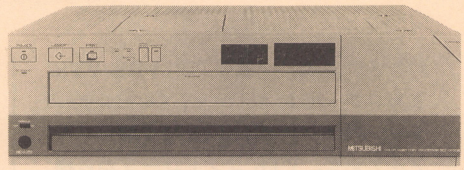
三菱電機は，フルカラーA4サイズの高解像度ハードコピーとしてカラー・ビデオコピー・プロセッサ「SCT-CP200」の販売を開始する。

同製品は，すでに販売されているA6サイズのハードコピーSCT-CP100の上位機種として位置付けられ，A4サイズの紙に出力できる。入力信号はパーソナルコンピュータなどのアナログRGB信号の他，NTSC，S-VHSなどに対応しており，水平周波数が15～36kHzの間の信号であれば自動的に追従するオートスキャン方式を採用した。

印刷方法として昇華式熱転写方式を採用したため，水平1,280，垂直960ドットの高解像度で出力する。これは，通常のアナログRGBディスプレイの解像度よりも高いが内部で色の補完を行っており，実際の画面上よりも色の表示がリアルなものとなる。表示色は，イエロー，シアン，マゼンタ各6ビット64階調の約26万色が出力できる。また，モノクロ出力も可能である。

入力は，パラレル・インタフェースを使用しているため，通常よりも高速なダウンロードができる。1枚当たりの印刷速度は約160秒となっている。出力の際には，日付，コメントなどを記入することも可能。専用のインクをカートリッジ交換でき，専用紙を含めた1枚当たりのランニング・コストは200円程度である。

価格は69万8,000円である。



問合せ先：三菱電機株式会社　電子商品事業部
〒100 東京都千代田区丸の内2-2-3
☎03(218)3108

土壌利用のための地図情報システム

社会調査研究所

日本土壌協会は，社会調査研究所が開発した土壌データ利用のための「要因別分布図作成システム」の販売を開始した。

同システムは，土壌，気象などのデータ項目ごとの分級図を作成するソフトウエアであり，特定の作物の成育適地や各種土壌要因別の疎外要因を判別することにより，作物への影響が明らかになり最適な判断を行える。

システムは，データ管理部，検索部，出力部から構成されており，データ管理部はデータベースの登録・更新を行う。検索は，市町村や図幅単位など地域検索と土壌の種類，気象条件など条件検索もできる。出力は，メッシュ，ポリゴン，グラフなどの画像と統計表などの帳表が出力できる。

動作環境はPC-9801シリーズのMS-DOSであり，システムには，土壌図データ，示性分級式，ネームファイルなどの基本データが含まれている。

価格は20万円となっている。

問合せ先：財団法人日本土壌協会
〒101 東京都千代田区神田神保町1-58
☎03(292)7281

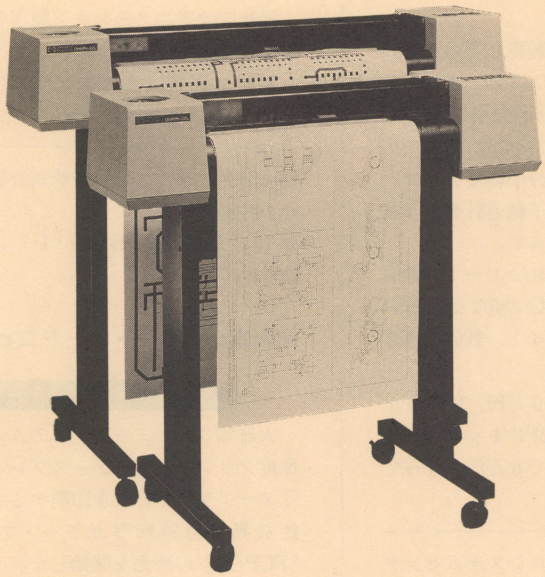
パーソナルCAD用 大型ペン・プロッタ

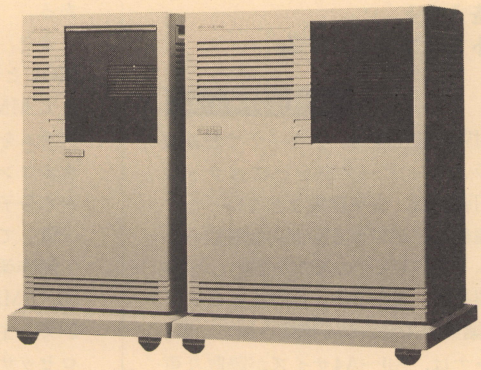
YHP

横河・ヒューレット・パッカードは，パーソナルCAD用の大型プロッタ「DraftProDXL(HP7575A)」と「DraftProEXL(HP7576A)」の販売を開始した。

今回発表された2機種は低価格化が進むパーソナルCADシステムに対応したもので，従来機種に比べ高精度で低価格

new products





を実現した。作画はマイクログリッド方式を採用し，精度は機械的分解能0.0125mm，再現性0.1mmの高精度，出力は加速度2.8G，80cm/秒の速度で行うことができる。

また，米国ヒューレット・パッカードのグラフィック言語であるHP-GLを標準内蔵しているため，従来機種との互換性が保たれている。オプションで1Mバイト，2Mバイトのバッファ・メモリ，漢字ROMなどが準備されている。

価格は，A1～A4サイズ対応のDXLが95万円，A0～A4サイズ対応のEXLが120万円となっている。また，オプションの1Mバイトのバッファ・メモリが13万5,000円，2Mバイトが23万5,000円，漢字ROM＋HP-IBインタフェースが10万8,000円となっている。

問合せ先：横河・ヒューレット・パッカード株式会社　コンピュータ第2部門周辺機器マーケティング
〒168 東京都杉並区高井戸東3-29-21
☎03(335)8234

VAXファミリー新製品4機種

日本DEC

日本ディジタル イクイップメントは，VAXシリーズの新機種として，汎用小型コンピュータ「MicroVAX3300」「MicroVAX3400」と，これらをベースとしたネットワーク・サーバー「VAXserver 3300」「VAXserver3400」の販売を開始した。

MicroVAX3300/3400は，CMOS技術によりLSIを1チップ化することでMicroVAX IIに比べほぼ同じ価格で約2.5倍のCPU性能を実現した。メモリ容量は，4Mバイトのオンボード EECメモリと8Mバイトの EECメモリ・アレイ・モジュールの計12Mバイトを標準装備，8Mバイトずつ最大28Mバイトまで拡張できる。

固定ディスク装置として，1基150Mバイトの新開発のインテグレーティッド・ストレージ・エレメント(ISE)RF30により，MicroVAX3400で最大900Mバイトまで搭載可能である。また，新しいデジタル・ストレージ・システム・インターコネクト(DSSI)と組み合わせることによって複数のISEが同時に動作でき，従来の機種に比べてデータ転送速度が向上した。

OSは，基本のVMS，VAX版UNIXであるULTRIX-32，リアルタイム専用のVAXELNの3種類が使用でき，従来のVAXファミリーのアプリケーション・ソフトウエアはそのまま使用できる。I/Oバスは従来と同じQバスを使用している。

同社では，オフィス，流通サービス，金融，研究開発，製造などさまざまな分野での需要を見込んでいる。

VAXserver3300/3400は，それぞれMicroVAX3300/3400をベースとしたネットワーク・サーバーであり，LAN接続されたEWSやパーソナルコンピュータのプログラムやデータの管理への需要を見込んでいる。

価格は，MicroVAX3300が970万円から，MicroVAX3400が1,172万円から，VAXserver3300が591万円から，VAXserver3400が793万円からとなっている。

問合せ先：日本ディジタル イクイップメント株式会社 広報宣伝部
〒170 東京都豊島区東池袋3-1-1
☎03(989)7145

グレースケールも取り込める画像処理装置

ダイキン工業

ダイキン工業は，ビジョンシステム(汎用画像処理装置)「ROBOTEC V-200」の販売を開始した。

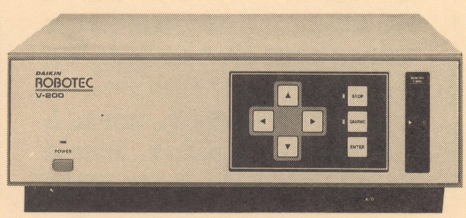
同製品は，従来は高価格機でしか行えなかった256階調のグレースケールの画像取込みが可能になるとともに，512×480ピクセルという同社従来比の2倍の取込みが可能になった。これにより，二値化システムでは得られなかった鮮明な画像が得られる。

また，専用CPUを採用した高速化，日本語表示メニュー，BASICライクな高級言語の採用によるユーザー・カスタマイズの容易さ，外部記憶装置としてメモリカードを採用する，などの特徴がある。

インタフェースは，セントロニクス，RS-232C，RS-422，デジタル入出力などに対応している。

価格は170万円となっている。

問合せ先：ダイキン工業株式会社　ロボットシステム部
〒566　大阪府摂津市西一津屋1-1
☎06(349)7361



new products

図面を入力しベクトル化するシステム

中村留精密工業

中村留精密工業は，スキャナで読み取ったラスターデータをもとにベクトル化，円弧認識，文字認識などを行ったうえで編集などができる図面自動読取りシステム「POLUX」の販売を開始した。

同製品は，スキャナ，ベクトル化システム，編集システムから構成されている。スキャナは，A0～A4サイズが読取り可能であり，解像度は最大16ドット/mm，A0判の読取り時間は80秒となっている。ベクトル化システム部では，ノイズ除去を行ったうえで芯線化，ベルトル化の順に行い，円弧，文字，線などを認識する。編集システムは，ベクトルデータの解析と円弧，文字，線などの追加，変更，削除などの編集作業ができる。また，オプションで，汎用CADに入力ができるようにIGESフォーマットへの変換を行うソフトウエアが準備されている。

システムは，ハードウエアにPC-98XL²(主記憶2.5Mバイト，数値演算プロセッサ付き)とイメージスキャナTOMEC-EYE，それにベクトル化ソフトウエアと編集ソフトウエアの構成で985万円となっている。

問合せ先：中村留精密工業株式会社
OA機器事業部
〒920-21 石川県鶴来町鶴野町口15
☎07619(3)1111

パソコン用分子計算ソフトウエア

東レシステムセンター

東レシステムセンターは，本格的なパーソナルコンピュータ用分子計算ソフトウエア「パソコンMOPAC」の販売を開始した。

同製品は，パーソナルコンピュータで初めて水素原子20，水素以外の原子11について電子状態の計算などの分子軌道計算を行うことができるソフトウエアであり，分子軌道計算で世界的に有名なプログラムMOPAC(Ver.3.1)をパーソナルコンピュータに移植したものである。遷移状態計算，グリッド計算，多重項状態計算などのいくつかの機能を除いてMOPACの機能をそのまま継承しており，16ビット・パーソナルコンピュータで3次元分子計算が行える。MOPACから機能拡張されたものとして，計算の収束状態のリアルタイム・モニター機能がある。

また，同社がすでに販売している3D-MOLという分子モデリング・プログラムとMM2PPという分子力場計算プログラムを組み合わせることにより，パーソナルコンピュータ上で分子のモデリングから分子力場計算や分子軌道計算までを一貫して行うことができる。

動作環境は，PC-9801シリーズ，PS/55，FMRシリーズ，PC-286などとMS-DOS。主記憶640Mバイト，数値演算プロセッサが必要である。

価格は，MOPACが9万円，3D-MOLが18万8,000円，MM2PPキットがパーソナルコンピュータ用で6万円となっている。

問合せ先：株式会社東レシステムセンター　営業部
〒103 東京都中央区日本橋本町1-5-9
☎03(245)5006

大容量高速画像メモリ

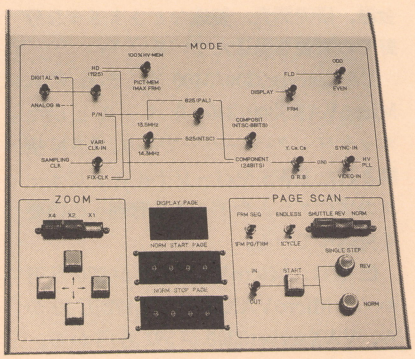
ぺんてる電子

ぺんてる電子は，ランダムに画像データをアクセスすることが可能な画像メモリ装置「MFR-IK型」の販売を開始した。

同製品は，最大7Gバイトの画像メモリを使用して，画像データをランダムかつリアルタイムに取り出すことができる装置であり，従来この種の製品ではディスク装置を使用していたものである。

入出力ともNTSC，PAL，HD-TV，RGBに対応しており，フレーム数はNTSCコンポジット時で最大3,840まで格納することができる。入力はリアルタイムで行え，通常のビデオ入力の他スロースキャン・デジタルと映像データの入力が可能となっている。

カラー表示の場合R,G,B各8ビットの1,670万色，モノクロ表示の場合256階調での表示が可能である。マルチバス仕様のコンピュータに対応している。



同社ではアニメーション製作などへの需要を見込んでいる。

価格は実装メモリに応じて2,000万～7,000万円となっている。

問合せ先：ぺんてる電子株式会社　システム技術営業部
〒104　東京都中央区新川1-7-11
☎03(553)6549

高画質のカラー・ハードコピー

大日本コンピュータシステム

大日本コンピュータシステムは，高解像度グラフィック・ディスプレイの画面イメージを，そのまま印刷することが可能な昇華型熱転写カラー・プリンタ「FCP-10」の販売を開始した。

同製品は，最近のグラフィック・ディスプレイの高解像化，多色表示化に対応したもので，画面イメージを写真と変わらぬ画質で再現することができる。

100MHzと60MHzのビデオ・インタフェースを内蔵しているため，ホスト・コンピュータを介さず直接グラフィック・ディスプレイと接続でき，ホストマシンに負担がかからない。

表示方式は昇華型熱転写方式を採用，表示色はR，G，B各128色の約210万色。分解能は10ドット/mm，出力サイズはA4判の専用紙またはOHP用紙において1枚当たり最大3分で出力する。

接続可能ハードウエアは，IBM5080，IRIS，Sun，COMTECなど各種EWSに対応している。

価格は650万円となっている。

問合せ先：大日本コンピュータシステム株式会社　サイエンス営業部
〒160　東京都新宿区高田馬場2-14-2
☎03(207)3321

自動NCデータ作成システム

コムシス

コムシスは，スキャナで読み込んだ画像に対してグラフィック処理を行ったうえでベクタライズし，NCデータを作成するシステム「C.C.N.システム」の販売を開始した。

従来，NCデータを入力するにはデジタイザでトレースしてデザイン・ベクトル化を行っていたが，同システムでは，ラスターデータ→CGデータ→CADデータ→NCデータの一連の過程を自動化することによって作業の省力化を図っ

new products

た。また，入力したデータは各セクションで編集できる。

作業例として，まず，スキャナで読み込んだ画像に対して色・デザインの修正の後ベクタライズし，カラー・グラフィックの色ごとの輪郭線を抽出しCADデータを作成，この段階で文字や他のデザインとの合成を行う。次に，運針情報などNCデータの各種設定を行ったうえでNCデータを作成する。画像データの編集は，ラスターデータ，ベクトルデータそれぞれに対して行うことができる。

グラフィック解像度は16,000×16,000ドットで，1,677万色中256色の表示が可能である。ラスターデータはカラープリンタに，ベクトルデータはX-Yプロッタにそれぞれ処理の途中で出力可能である。

システムは，ホスト・コンピュータにPC-98XL²，カラースキャナ，ディスプレイ，カラーインクジェット・プリンタ，デジタイザ，画像コントローラ，それにソフトウエアから構成されている。

予定価格は500万〜600万円。

問合せ先：株式会社コムシス
〒101 東京都千代田区岩本町3-5-12
☎03(865)4591

高速サーマル・プロッタとレーザー・プロッタ

マックス

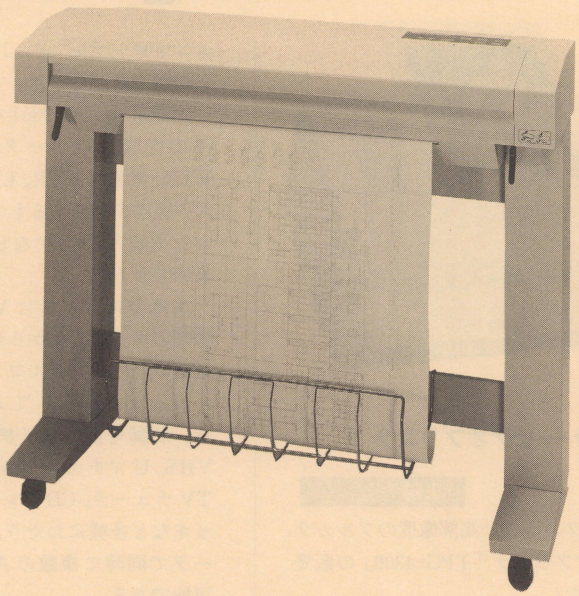
マックスは，感熱方式のプロッタ2機種とレーザー方式プロッタ1機種のテスト販売を開始した。感熱方式のRP-600TはA1判で静電プロッタの約1/2の低価格を実現している。RP-900Tも感熱方式で，A0判の出力が可能。レーザー方式のRP-300LはA3判の図面を約6秒で高速出力することができる。また，3機種ともリピート機能を備えており，同じ図面を99枚まで連続出力することも可能である。

RP-600T/900Tの印字解像度は約200dpi，A1判の出力時間は約60秒となっている。給紙に関しては，専用ロール紙1巻でA1判図面100枚まで出力することが可能で，RP-600Tの場合には自動的に図面を切るオートカット方式が採用されている。

RP-300Lはレーザー方式によって普通紙に約400dpiの印字解像度で出力することができる。用紙はカセット給紙方式によって250枚までを一度にセットすることができるため，用紙を取り替える手間を省くことができる。

価格は，RP-600Tが160万円，RP-900Tが180万円，RP-300Lが150万円である。

問合せ先：マックス株式会社　図形事業部
〒103　東京都中央区日本橋箱崎町6-6
☎03(669)8100



UNIX専用EWS

精工舎

精工舎は，UNIXベースのEWS「vision386」の販売を開始した。

同製品は，CPUに25MHzのi80386，数値演算プロセッサにi80387を採用し，OSにUNIXベースのTechno-UX3.0を採用したEWSであり，CAD/CAM/CAE，ソフトウエア開発，グラフィックス処理，論理シミュレーション，数値解析などへの需要を見込んでいるとともに，今後EWS市場に積極的に参加していく予定である。

ネットワークはRFSをサポートしており，同じシステム間で資源の共有ができる。漢字入力はJAE環境下で使用することができ，内部コードはUNIX標準のEUCとなっている。グラフィック・ライブラリとしてGKSがサポートされている。X-Window，Ethernet，NFSもサポートする予定である。プログラム言語としてC，FORTRANがサポートされており，数値演算プロセッサとしてWTL3167も使用できる。

今回発表されたのは，固定ディスクが86Mバイトの「vision386/10」と320Mバイトの「vision386/30」の2機種。

価格は，vision386/10が395万円，vision386/30が490万円となっている。

問合せ先：株式会社精工舎　情報機器事業本部第1販売部システム2課
〒130 東京都墨田区大平4-1-1
☎03(621)4011

NCツールパス・テスト用ソフト

ニチメン

ニチメンは，米コンピュータ・グラフィック・テクノロジー社のNCツールパス・テスト用アプリケーション・ソフトウエア「VERICUT」の販売を開始する。

これまでCAD/CAMシステムや自動プログラミングによってNCツールパスを作成した場合，モデル（被切削物）の最終形状のチェックは，工業機械によって何度も繰り返していた。同製品はこの試し削りをCGによってシミュレーションするため，手間と材料が少なくてすむ。

操作は，マウスとキーボードにより対話型で行い，データ形式は多くのCAD/CAMで使用されているAPT仕様のCL（Cutter Location）であるためデータの再利用ができる。表示は3次元ソリット・モデルで行い，モデルを削る様子，切削後のモデルの形状などの一連の様子を表示することができる。

予定価格は120万〜130万円である。

問合せ先：ニチメン株式会社　電子情報機器部情報機器課
〒103　東京都中央区日本橋3-11-1
☎03(277)5083

new products

フルカラー・ビデオプリンタ

東陽テクニカ

東陽テクニカは，高解像度のフルカラー・ビデオプリンタ「TPG-4300」の販売を開始した。

同製品は，同社従来機種の約1.5倍の記録密度である300dpiと，熱転写プリンタでは最高の解像度を実現するとともに，広帯域インテリジェントRGBビデオプロセッサを標準装備することによりフルカラーを実現した。

トリミング，印刷位置の指定，自動センタリング，白黒反転，カラーアサイン，色強度調整，YMC個別プリント，最高99枚の自動コピー，左右反転，90度ローテーションなどの各種機能の他，0.1倍単位で指定できる最高16倍の拡大機能はX，Yそれぞれの方向に指定できる。取込み画像は1,280×1,280の解像度に対応しており，約2秒で読取りを行う。

価格は198万円となっている。

問合せ先：株式会社東陽テクニカ　貿易部CVP販売
〒103 東京都中央区日本橋本石町1-1-2
☎03(245)1351

マルチAVコントローラー

ケーシー

ケーシーは，コンピュータからAV機器を制御することができる「マルチAVコントローラー」と，16種類の制御ボードの販売を開始した。

同製品は，PC-9801などのコンピュータで作成したデータに従い，家庭用VTR，業務用VTR，LDP，CDなど各種AV機器を制御するもので，マルチビジョンや館内CATVなどの映像を無人で表示できる。

本体のスロットにVTRに応じた26種類のボードの中から選択したものを差し込み，パーソナルコンピュータなどをホストマシンとしてAV機器を制御する。制御可能なAV機器は，ベータ，VHS，Uマチック，8mmビデオ，VHD，TVチューナ，CDプレーヤ，カセットデッキなど多種にわたり，1台のコンピュータで同時に複数のAV機器の制御が可能である。

制御はRS-232Cで行うため，多数のコンピュータに対応している。今回発表されたボードは17種類で，パラレルI/Oボードやセントロニクス・ポートから受信するボードなどがあり，最大20チャネルに対応している。

同社では同製品の発売と併せて，コンピュータ制御データのプログラミング，映像演出の企画，CG・ビデオ映像の制作，システムコンサルティングなどのサポートも行っている。

また同社では，博物館やショールームなどへの需要を見込んでいる。

価格は，2UタイプのAVC-2Uが19万円，4UタイプのAVC-4Uが23万円，ボードが3万6,000～7万9,000円，標準的なシステム構成が40万円前後，20チャネル・マルチプレクサ仕様で70万円前後となっている。

問合せ先：株式会社ケーシー
〒550　大阪市西区北堀江1-1-7
☎06(535)4339

低価格の大型プロッタ

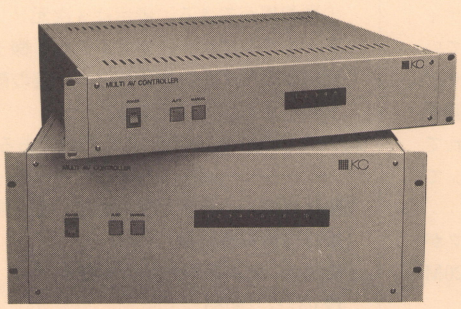
ミマキエンジニアリング

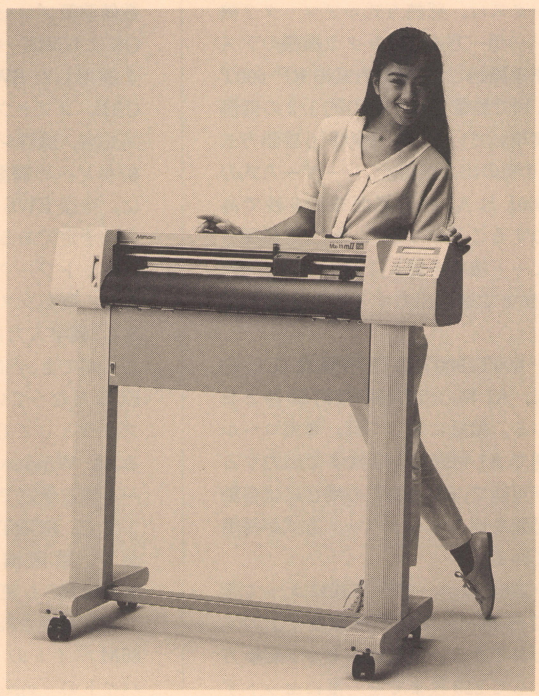
ミマキエンジニアリングは，大型，高速，低価格のデジタル・ペンプロッタ「北斎シリーズMX-11m II」の販売を開始した。

MX-11m IIは，従来販売されているMX-10/11シリーズの低価格版で，作図サイズはA1判のデジタル・ペンプロッタである。作図精度は距離精度が0.5mm，直角精度が0.6mmの高精度で，作図時間は60cm/秒の高速を実現した。また，28K～540Kバイトの受信バッファを内蔵しているため，ホスト・コンピュータへの負担が少なくてすむ。出力用紙のサイズは最大A4判で，A1～A4判の入力用紙のサイズを自動検出する。

価格は98万円となっている。

問合せ先：株式会社ミマキエンジニアリング 営業部
〒150 東京都渋谷区恵比寿南1-20-11
☎03(760)5500





変わることにより高め合える，交わらなければ高め合えない。

From The Editor's Desk

日本人はコミュニケーションが下手だ，友達のつくり方が下手だと多くの外国人が指摘する。理由はいろいろと考えられるが，指摘はおおかた正しいであろう。例えば，コンピュータ・グラフィックス分野の経営者同士の付合い，CGデザイナー同士の付合いなどは，米国と比較して格段に少ない。フランクに付き合う機会が少ないということだけではなく，明らかに避けている，または必要がないと判断しているのではないかと思われる。

付合いとは雑談をする，お酒を飲み合うということだけにとどまるのではなく，コミュニケーションを深めることにより互いに啓発し合い高め合う，またはビジネスとして良いパートナーになれるかどうかを考えるといった意味のことである。

同じような仕事をしている者同士が話し合うことで，技術や経営ノウハウを盗まれる，何か損をするという被害者意識ばかりが先にでてしまい，自分が高められる，得をするという意識がわが日本人には少ないようだ。また，外国人とは積極的に接していても日本人には警戒するというエセ思考の人も見受ける。

この号のトレンド記事の中でステラ社のポダスカ氏が述べているように，以前の会社とは依然として個人的にも企業的にも良い関係をもっている，社員が良いアイデアをもってスピンアウトする場合は彼の成功を願っている――といった考えは日本人の平均的な思考ではないだろう。もちろん，フランクに話し合える関係をもっているからといって自社の企業秘密的なことまで漏らすというわけではない。あくまでも企業人として企業の利益を追求することが前提であることに変わりはない。

高い意識をもっている人であれば容易に納得していただけることと考える。

1989年2月1日

河内隆幸

メタボールと私

META EDITOR開発の経緯と筆者が使用してきたシステムの変遷，メタボールの造形的特性と医学分野などへの応用例と可能性について述べる。

高沖　英二*

はじめに

私自身がいまだにコンピュータ・グラフィックス(CG)の初心者であると思っているのに，メタボールについて書くことは真にせんえつではあるが，ここ2，3年メタボールにかかわってきた者としての経験と，その経験を通して抱いた多少の感想を述べてみたい。自分が思っていることをCGで表現してみたいと思われる方や，メタボールに関心のある方の参考となれば幸いである。

まず，私がなぜ，どのようにCG—特にメタボールにかかわり始めたのか，そして今や手放せない道具となったMETA EDITORがなぜできたかという辺りから話を始め，次にメタボールに関する私見を，さらにメタボールの応用についての展望を述べる。

META EDITORの開発とシステムの変遷

■メタボールに出合う

私のCGにおける活動は，もともとメタボールで何ができるだろうかという発想からではなく，CGアニメーションで本当に美しい人体に本当に美しい動きをさせたい，という欲求から出発している。

以前から彫刻を勉強していた私は，ある日突然CGアニメーション作品をつくることを思い立ち，当時大阪大学におられた大村皓一先生の門を叩いて，1986年の春から1年間電子工学科の研究生にしていただいた。当然，生活のために働かなければならず，仕事の合間にときどき研究室に顔を出すというような状況だった。

メタボールとMETA EDITORを用いた作品例(COLOR IMAGESも参照)

C言語を学び始めて最初に作ったプログラムは，対数らせん関数を使って貝の形のポリゴンデータをはき出すツール(「作り貝」と名付けた)で，貝殻の縁の形状を与えると巻貝でも二枚貝でもできるというものだった(「まずカイより始めよ」などと冗談を言ったりした)。それを使ってホタテ貝を作ったのだが，その波打った面が十分滑らかになるようにポイントを細かく取るとデータが巨大になる。パッチではデータが膨大になってしまう，という印象をそのときにもった。もともと人体彫刻を学んできた者にとって，微妙な曲面やその微妙な輪郭が不本意に，少しでも角張ってしまうことは耐えられないのである。「CGによる人体表現」が研究テーマであったので，当面パッチは使えないという結論をだしてしまったのである。

もちろん新しいオブジェクト・モデリング法を開発すること

*たかおき　えいじ　㈱メタ・コーポレーション・ジャパン　〒150　東京都渋谷区神山町1-3-103

も考えたが，とにかく早くアニメーションをつくりたいという一心で，とりあえずメタボールを使うことにしたのである。しかし，メタボールのモデリングは試行錯誤を必要とするため，LINKS-1システムの大部分を長時間占有することになる。当時大村先生のもとには多くの学生，大学院生，それにアーチストが集まり，大変混み合っていたので，ほとんどそういうことが許される状態ではなかった。そこで私は，パーソナルコンピュータ上でメタボールのモデリングもできるエディタをMETAC（メタック）という名で開発することにしたのである。

METACは当初から単にメタ専用のモデラーではなく，アーチストが自由にその創造性を発揮する環境として考えていた。アーチストが何かをしようと思えばすぐにそれができるように，ということを常に考えながら開発を行ったのである。例えば，マウスの機能切替えの高速化のために，オクティコン(OCTICON)という新しいメニュー方式も開発した。このメニューは，呼び出されるとカーソルを中心に放射状に並んだ8個のアイコンがポップアップし，マウス・カーソルのスライドする距離ではなく，その方向によって選択される。したがって，わずかな動きで素早く，しかも覚えてしまえば全く見なくても使えるのである（ブラインド・マウス）。このオクティコンは，特に試行錯誤が必要なメタボールのモデリングに効果を発揮している。

METAC開発には当初の予想以上に時間がかかり，メタボールのモデラーとして一応使えるようになったのは1986年の11月ころだった。METAC自体はまだ完成していないが，META EDITORはそのメタボール・モデラーの機能を独立させたものである。そのころに医学部の辰巳治之氏と共同で臓器の組織の連続切片から3次元像を再構築する研究を始めたのだが，これについては後でもう少し詳しく述べる。

その年の暮れからいよいよMETA EDITORを使って人間の顔を作り始めたが，年を越してからもCG以外の仕事に追われ，ほとんどLINKS-1でレンダリングをしないうちに研究生としての1年間が終わってしまった。

■パソコン・マルチタスクの時代

1987年4月から，トーヨーリンクス（現在，リンクス）よりお借りしたPC-98版PERSONAL LINKS上でようやく本格的な画像生成が始まった。**図1**(COLOR IMAGES参照)は1986年の5月ころ制作したもので，PC-9801Eを用い，レンダリングに約10時間かかっている。ちょうど，夜にデータをセットしておけば翌朝に絵が出来上がっているといった具合である。

私が使っていたシステムにはいろいろ変遷はあったが，最も充実した時期には，中古で買った4台のPC-9801シリーズ（うち2台はディスクレス）を1台のハードディスク・ドライブ（ランドコンピュータ製LHDU40）につなぎファイルを共有しながら，1台のCPU上でMETA EDITORを使いモデリング，2台でそれぞれ別のデータを与えてレンダリング，残りの1台でMETA EDITORの開発を並行して行っていた。ディスプレイの1つは切替えスイッチをつけて2台のCPUで共有していたが，キーボードは4台を2段にして並べていた。1人でこのシステムを操っていたわけだが，実際にMETA EDITORを使いながら，改善点やバグに気がつくとすぐに隣の開発用CPUでプログラムを書き換えコンパイルをかけて，またモデリングに戻る。そうこうしているうちにレンダリングが完了するといった具合で，大変能率が上がった。ただ，泣きどころは主記憶容量の不足で大きなデータがのらない，つまり人間の全身のデータが一度にレンダリングできない，という点だった。

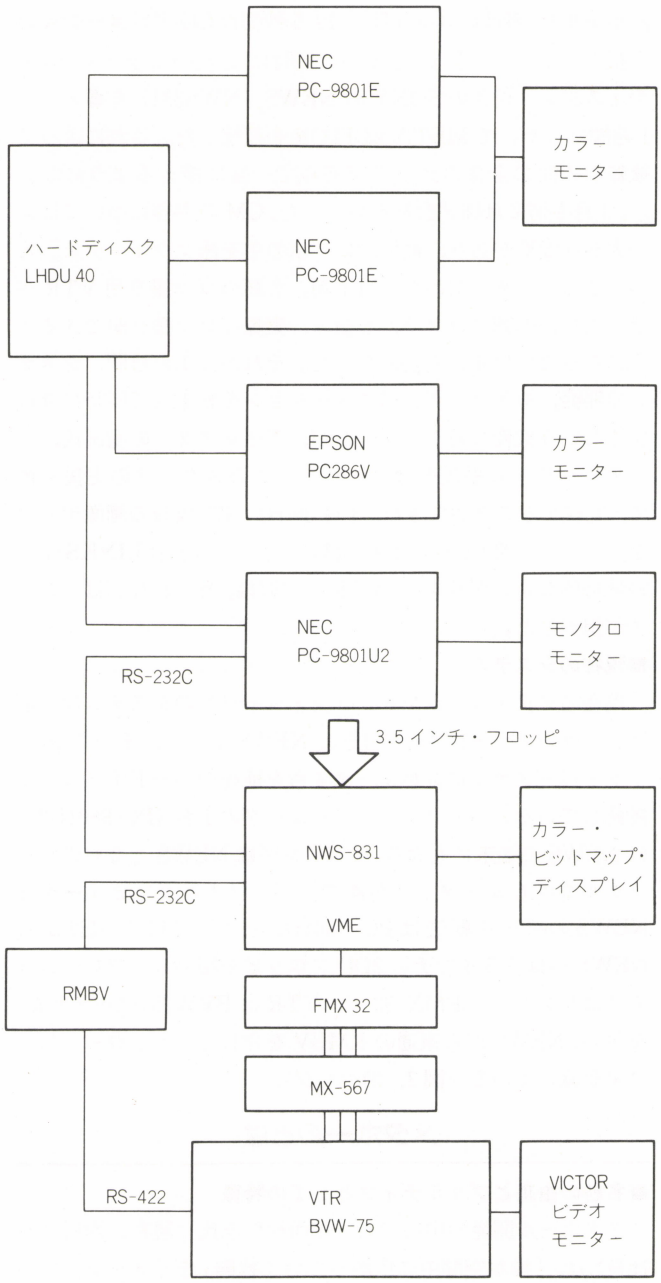


図2　当社のシステム構成図

1987年の9月に活動の場を東京に移したが，その契機はNHKの伊藤博文氏との出会いであった。彼はいち早くMETA EDITORに注目し，NHKエンタープライズの「人体」プロジェクトのCGチームでこのソフトウエアを活用している。

■会社を作り，ついにアニメーションが可能になった

1988年2月になってIMAGICAのテレビCMの制作依頼の

話が決まり，最後のロゴを除く12.5秒間のCGアニメーションを担当することになった。それと同時にカラー・ビットマップ・ディスプレイ付きのSONYのNEWS（NWS831）を導入し，1週間ほどかけてMETA EDITORを移植した。これによって複雑な人間の全身のデータが初めて一度に扱えるようになった。3月中旬には株式会社を設立した。CMの内容については途中大きな変更があり，最終的に「水の中を泳ぐヴィーナス」ということに決まったのが3月下旬。念願のコマ撮り用VTRが手に入り，制御プログラムを組み，実際にコマ撮りができるようになったのは4月の初めだった。それから1カ月間，システムの開発，モデリング，アニメーションを私1人で同時に進行するという大変な日々を送ったが，ディレクターの増田氏，プロデューサーの平氏をはじめリンクスのスタッフの支援を得て，完成することができた。4月30日に375枚分の画像データをリンクスに渡し，私の仕事は終わった。その後のLINKS-1での最終的なレンダリング，VTRへの収録，音入れなどはリンクスが担当した。

■現在のシステム

現在のメタ・コーポレーション・ジャパンのシステムは，前述のカラー・ビットマップ付きのNEWSが1台，それに16ビット・パーソナルコンピュータ4台を前述のハードディスクに接続している。パーソナルコンピュータの1台（PC-9801U2）はNEWSの端末にもなる。モデリングはNEWSと2台のパーソナルコンピュータ上で可能で，パーソナルコンピュータとNEWSのデータ転送はPC-9801U2を介して行う（U2からNEWSへは3.5インチの2DDフロッピを用いる）。フレームメモリはリンクスのFMX-32で，VTRはBVW75（ベータカム）を用い，NEWSから東通のRMBVを介してコントロールしてコマ収録している（**図2**，前ページ）。

メタボールとは

■名前の由来とプリミティブとしての特徴

メタボール開発の中心であった西村仁志氏（現在，西村技研社長）は，「煙が空間中に広がっていく状態」をイメージしていたそうだ。彼によるとメタボールという名称は，当時開発中の濃度分布関数によるプリミティブ群をメタモルフォーシスという意味でメタと総称していたので，ボール状のものはメタボールとなったということである。なおTRACYではメタというプリミティブにメタボール（ball）とメタ楕円体（ellipsoid）がある。今まで単にメタボールという言葉を使ってきたが，実際にはメタ楕円体を指している。

メタボールのアルゴリズムについてここではふれないが，用語を簡単に説明しておく。

メタの重さとはボールの中心部の濃度であり，濃度は濃度分布関数に従って周辺部へいくほど低下し，ついに0になったところが有効半径である。濃度＜（閾値＝1）となるところからみて濃度＝（閾値＝1）になるところに面ができる。1つのメタボール（球）に必要なパラメータは中心座標（x,y,z）と半径，重さの5個で，メタ楕円体は中心座標（x,y,z），軸の長さ（x,y,z），回転角（x,y,z），そして重さの10個である。

META EDITORにおいては，上に述べた概念と用語を完全に無視してモデリングすることができる。

■造形的にみて，メタボールはメタボール

メタボールは他のプリミティブとは異なり，日常的な感覚で何かに比喩することが難しい。そこがメタボールの面白いところである。確かにメタボールは何かの代替物ではない。結局，メタボールはメタボールというべきで，人類がここにきて初めて手に入れた新しい造形表現の手段なのである。そのような意味でも，メタボールには他のプリミティブにない魅力がある。

同じ粘土でも彫刻家によって全くとらえ方が違う。私の場合，前から押すと横にはみ出す，ということが粘土の魅力ではなく，自由に削り取ることができ，しかもやり直すことができる，ということがその魅力であり本質なのである。そういう意味で，私にとってメタボールは粘土とは本質的に異なるものであり，異なる魅力をもった素材なのである。

何年も彫刻を学んできた私にとって粘土で塑像を作るのは簡単なことなので，塑像や石膏像から3次元デジタイザでデータを取ってもよさそうなものだが，それをする気は今のところない。一つにはデジタイザの精度の問題であり，サンプリング・ポイントの数と手間，さらに最終的に望みの形がレンダリングされるかどうかの確証がないからである。

彫刻の世界では，はるか昔から「星取り法」というのがある。これは主に石彫を作るときに使う手法で，まず粘土で原型を作り，石膏でそのコピーを作る。次に石膏像の表面に多くの点（星）を取り，デジタイザの代わりに星取り器のスライド式の針の先を基準として目的の石を削っていくのである。この作業は人に任せてもよいのだが，最終的には彫刻家が自ら削らなければならない。それほど彫刻作品の曲面は微妙なものなのである。

■メタボールによる造形のこつ

メタボールの使い方は人それぞれでよいわけで，これという決まった方法はない。適当に投げ込んでいけばそれなりの面白い形ができるものである。しかし，いつも偶然に任せていてはいずれ飽きがくる。どうしてもこういう形が表現したいという気持ちが起こってくるのが自然である。

ここでは，何か作りたい形が決まっていてそれが頭の中にあるか，または目の前にあるようなとき，メタボールを使ってどうすればその形をうまく再現できるかという問題について考えてみよう。

通常の美術教育でも「最初に大まかに形をとらえるように」と教えることが多いが，私はメタボールの造形においてもこの考え方が通用すると考えている。実際に私自身もほとんどの場合そういう作り方をしている。できるだけ少数のメタボールでその形の本質を表現したいという気持ちがある。芸術表現において，無駄や無意味というものは往々にして嫌われる傾向にあり，私もそういうものは避けたいと考えている。無意味なメタボールをやたら多く置きまくるより，要点を押さえるようにメタボールを1つ1つ，最も効果的なところに決めていくのはなかなか気分の良いものである。このように芸術表現上の要求が計算時間を減らすというのは，はなはだ結構なことである。

まず形を大きくとらえる。すなわち，大きなメタボールを使

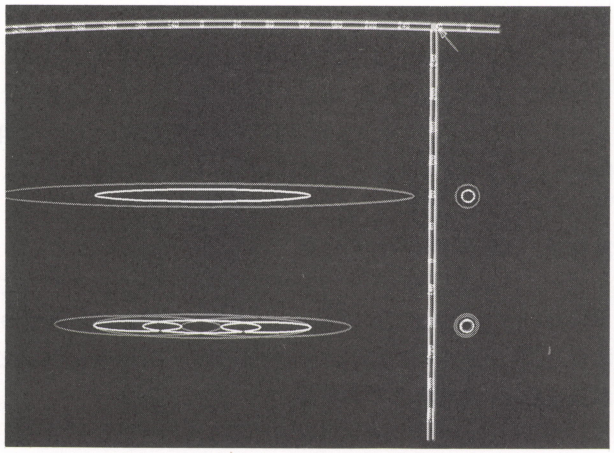


図　3

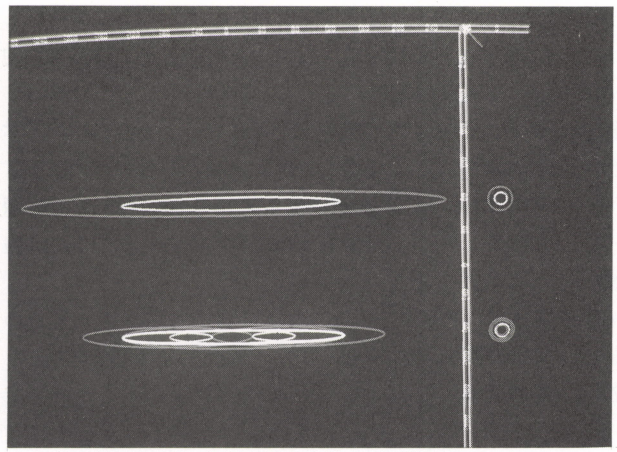


図　4

◉ 図 3〜5 は COLOR IMAGES も参照。

って全体の大まかな形を再現する。例えば人体を作る場合，胴体を1つのメタボール，腰に1つ，脚を大腿，下腿，足部にそれぞれ1個ずつで表現する。同じように上腕，前腕，手をそれぞれ1個ずつ，そして頭部に1個，首に1個，合計16個のメタボールを使って最も大まかな人体を表現してみる。そして，重さを加減しながら融合の状態をみてみる。初心者はここで何度も試行錯誤するのがよい。何度も試行錯誤していると，どうしても1つのメタボールではまずい，というところが見えてくる。そうなったら，その1つのメタボールの代わりに2つのメタボールを置いてみる。おそらくこういうことの繰返しによって最もシンプルでけじめのある形ができてくるに違いない，と私は考えているのである。

■断面を見ることの重要性

META EDITOR には，メタボールでできた物体（つまり複数のメタボールどうしが融合した結果の形）の断面を表示する機能がある。この断面を連続的に取っていくと，その物体の形状がかなり正確に把握できる。それは，等高線によって地形が正確に把握できるのと同じことである。ある意味ではシェーディングした画像よりも，等高線表示の方が形状を正しく伝えているのである。断面の形を把握することは，物の形をより深く理解することにつながる。また見取り図，透視図よりも平行投影した3面図の上でモデリングした方が，正確な造形が可能となる。これらのことはすべて，見る方向あるいは見る方法をいろいろと変えてみることの重要さを示している。CG ではこの見方に多様性を与えるのが容易である。

■アニメーションのための配慮

例えば，真っすぐに伸ばした腕を作ってよくできたと思っていても，肱を折り曲げると関節部が不自然に変形してしまうということが起こる。アニメーションにする場合は，あらかじめ関節などの運動の範囲を考慮して造形する必要がある。**図 3** に見るように，細長い楕円体は有効半径が意外に長軸方向へ伸びているので注意が必要である。このような場合には**図 4** のように，そのメタボールを3つのメタボールに置き換えることによって長軸方向への広がりを減らすことができる。

諸分野への応用

■医学教育

現在，私の唯一の弟子である当社の皆川氏は東京医科歯科大学第一解剖学教室の専攻生でもあるのだが，もともと医学・生物学の専門書のイラストを描くバイオ・メディカル・イラストレーターとして出版業界で仕事をしてきた。この仕事は肉眼では見えない世界のもの（細胞の中の様子，各器官が実際にどのように体の中に収まっているか，など）を，主に平面的な情報から立体的なリアルイラストに再構成するものである。そのようなノウハウがあったので，コンピュータに触れるのが初めてだったにもかかわらずすぐに META EDITOR を使いこなすようになったのだろう。**図 5** の骨格は彼女が META EDITOR

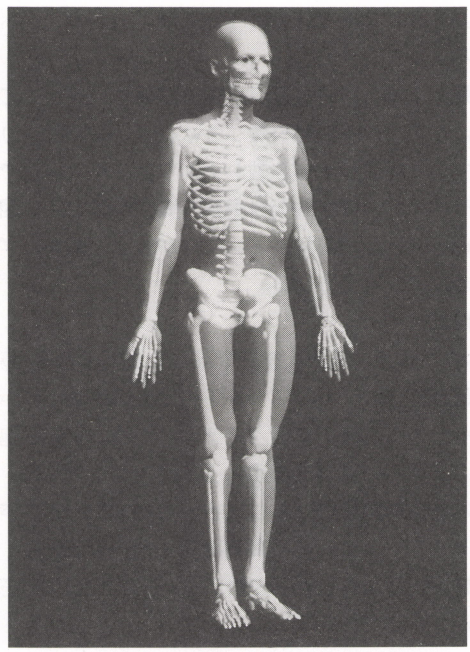


図 5　全身骨格

◉ 図 6，7 は COLOR IMAGES 参照。

を使い始めて正味約1カ月で仕上げたもので，数冊の解剖書といくつかの標本を参考にして，人の全身の骨1個1個の典型的な特徴をかなり正確に再現している。骨格というものはかなり個体差があり，ある1つの標本を忠実にコピーすることが医学教育のための教材として必ずしも最良とはいえない。最近，CTスキャンやNMRなど断層像による臓器や骨のCG化が話題になっているが，そこで再構築されたものはあくまでその被験者個人のものなのである。

CGの骨格は，アニメーションにして動きを与えたときその利用価値は飛躍的に増大する。今後，私たちは時間軸を加えた3次元動的解剖学とでもいうべきものを進めていこうと考えている。

では，実際の制作過程を脊柱（背骨）を例にとり紹介しよう。

モデリングはすべてパーソナルコンピュータ上で行った。まず胸椎の代表として第7胸椎（胸の辺りにある背骨）から作り始めた。脊椎骨はすべて左右対称なので，右側のみを作り鏡面コピーをした。META EDITORの等高線表示だけではなくレンダリングも数回行って形状を確認した。胸椎は全部で12個あるが，それらは基本的には同一の構造をもっているのでデータをコピーして作ることができる。つまり第7胸椎をコピーし変形させて第6胸椎，さらに5・4，8・9・10というように順次コピーと変形を繰り返していった。胸椎の基本形状と異なる頸椎（首の骨），腰椎（腰の骨）についても，それぞれの基本形を作りコピーし変形していった。仙骨は5個の骨が融合して1つになったものだが，今回は1つの骨としてモデリングした（すなわち1つのシェイプとした）。また尾骨は通常4〜5個あるが，これらを1つの骨として扱った。

パーソナルコンピュータ上で作った脊椎のデータをNEWSに送り，仙骨（腰椎を支える骨盤を形成する）を基本に，第5腰椎，第4腰椎と順に木構造を組み立てていった。脊椎は，どれも40個前後のメタボールでできているが，仙骨は約120個もある（**図6**）。全身の骨格では合計4,000個以上ものメタボールを使っている。

骨格を動かす場合，骨と骨が単純に関節でつながっているだけなので，メタボール特有の難しさはない（胸郭を除く）。メタボールでできた通常の人体を動かすよりはるかに容易である。関節にはそれぞれの標準的な自由度（専門用語では，関節可動域 range of motion（ROM）という）があり，動きのプログラムにはその数値を与えている。

■立体再構築

私は1986年より，大阪大学医学部第3解剖学教室（藤田尚男教授）の辰巳治之氏と，CGによる3次元再構築法について共同研究を行ってきた。**図7**は魚（タラ）の肝臓の伊東細胞（脂肪摂取細胞，ビタミンA貯蔵細胞，星細胞などともいう）および毛細血管の立体再構築像である（赤が毛細血管，紫色が伊東細胞の核）。メタボールによる立体再構築の利点は，細長く入り組んだ物体を扱うときにもデータ量が他の手法に比べて圧倒的に少ないということである。

電子顕微鏡の連続切片の像から再構築しようとする組織の輪郭を取り出して，その内側をメタボールで埋めていくのだが，将来，この作業を自動化することを考えている。

この研究は，昨年の3月31日〜4月3日に名古屋で行われた「日本解剖学会」と，11月30日〜12月2日の「札幌国際コンピュータ・グラフィックス・シンポジウム」で発表された。近年，解剖学，形態学の分野では立体再構築への関心が急速に高まりつつある。

■建築，工業デザインへの応用

例えば，ルイジ・コラーニ風の微妙な曲面をもったデザインを手軽に生み出すことができる。しかも，今後，模型の製作，木型，金型の製作まで一貫した自動化のシステムも可能である。

■美術教育の教材としての意義

前述したが，メタボールは適当に組み合わせただけで一見面白い形が簡単にできる。しかし，いろいろやっているうちに，良い形と良くない形，あるいは面白い形と面白くない形，美しい形と美しくない形，魅力的な形と魅力的でない形，強い形と弱い形，けじめのある形とけじめのない形などの差が見えてくる。このような生徒の試行錯誤の軌跡をそのままデータとして残し，再現できることは美術教育上，非常に有効である。

またメタボールをどう置くかという問題は，形をどのように理解し，解釈しているかという問題に他ならない。生徒がどのようにメタボールを置いていくかを見て，教師は容易に生徒の行った理解と解釈のあり方を把握することができる。

このようにメタボールは，美術教育のうえでも非常に有力な教材といえるのである。

おわりに

まず自然（人間を含む）を見る。次に古典を見て過去の人々が自然をどのようにとらえたかを学び，もう一度自然を見て自分のとらえ方を試みる。そしてそのとらえ方で表現してみる。この一連の作業を何度も繰り返すことが「芸術」だと考えて，私はそれを行ってきた。しかし「芸術」は，同時に社会に対する行為であるべきである。

以前の私の場合，表現の素材は粘土であり，石であった。自分で山へ行って粘土を取って来るわけではないが，自らの手で練り，自らの手で組み上げた支柱に粘土を付けていく。そして自らの手で削ったヘラで造形していくという，きわめて原始的な作業だった。石彫の場合も似たようなものである。

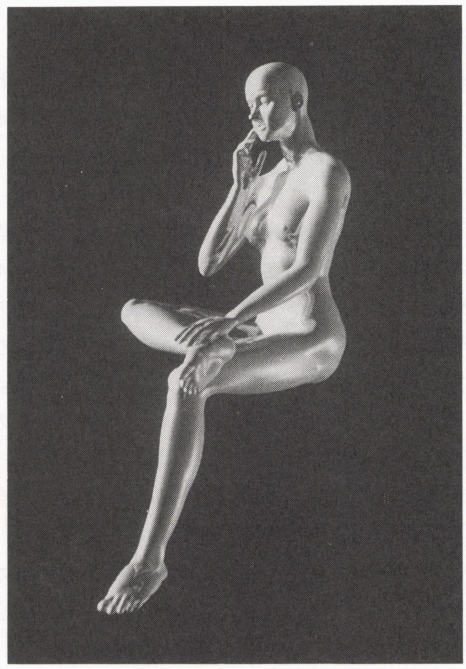
それに引き換えCGというものは，なんと多くの人々の知恵と情熱と努力のうえに成り立っているのだろう。私はたまたま必要に迫られてモデラーを開発したが，その開発に用する時間と労力は，ヘラを削ったりたがねに焼きを入れるときとは比べものにならないほど大きいことを知った。そのことを思うにつけ，CGにおいては「私の作品」といっても私1人の力でできたのではないということをひしひしと感じる。

CGはいろいろな意味で，従来の表現手段とは異なったものであるような気がする。もっと引き下がって考えてみれば，コンピュータという道具は従来の道具とは完全に異質なものなのかもしれない。私はそんなコンピュータの中に，十分に人間の美しさを持ち込んでおきたいのである。それがいつまでも真に人間のための道具であるように。

メタボール実験プログラム

実験的にメタボール（楕円メタ）を表示するプログラムを紹介する。

伊藤　博文*



◉図1，2はCOLOR IMAGES参照。

はじめに

1989年の6月から放送予定のNHKの大型企画番組「驚異の小宇宙・人体」では，各臓器や人体などの柔らかな物体を表現するためにメタボールを用いている。これまでメタボールは制御が難しいということで一般的ではなかったが，優れたモデリング・ツール（高沖英二氏の開発によるMETACなど）の登場により，従来のコンピュータ・グラフィックス（CG）では見られなかった高度なモデリングが可能になり注目を集めている。

人体CGプロジェクトではCGデザイナー金子徳明氏が胃腸，肝臓，心臓，胎児，人体などのモデリングを行い，メタボール専用のモーション・デザイン・ツールを開発してアニメーションを制作中である。

本稿では，プロジェクト内でメタボール関連のツールを開発するため，アルゴリズムの研究のために作られたメタボール（楕円メタ）を表示するルーチンを紹介する。

メタボールとは?

メタボールは大阪大学工学部において大村皓一氏（現大阪学院大学教授）を中心に開発された，実用的な濃度球アルゴリズムによるレイ・トレーシングのモデリングおよびレンダリング手法である。

通常のレイ・トレーシング・プログラムでは球，楕円体，三角錐，直方体などを組み合わせて，いわば積木のように形状を組み立てるという手法のため，臓器や人体のような滑らかな曲面を表現するには限界があった。また，曲面をパッチ（小さな三角形）を組み合わせて作る方法もあるが，人体のような複雑な曲面を覆い尽くすパッチはかなりの数になってしまう。また，輪郭の部分でパッチの1辺が直線的に見えてしまうので，完全に滑らかに見せるためにはさらに膨大な数のパッチが必要になる。

これに対して，いわば粘土のように造形するメタボールでは，比較的少ない数のプリミティブで複雑な曲面を表現できる。例えば**図1**の人体でも，せいぜい320個のメタボールしか使われていない。

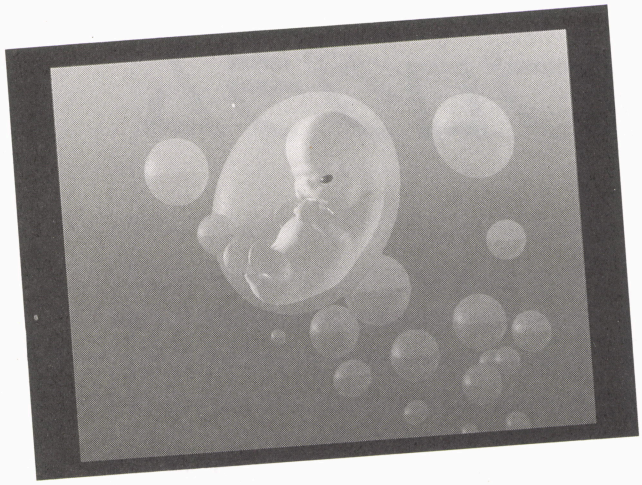
また，メタボールを用いると物体の滑らかなメタモルフォーゼが可能なので，人体プロジェクトでは胎児の発生のアニメーションなどに利用している（**図2**）。

メタボールの原理

濃度球は，空間内の点電荷分布の等電位面をモデルにすると理解しやすい。空間内にメタボールが存在するとすれば，その

*いとう　ひろふみ　㈱NHKエンタープライズ　人体プロジェクト　〒150　東京都渋谷区神山町4-14

◉図4は85ページ参照。



点にある一定の濃度＝電位（ポテンシャル）が与えられることになる。このメタボールの中心値（例えば1.5）はメタボールのウエイト（weight：重み）とよばれ，メタボールの中心から遠ざかるにつれて濃度は小さくなり，ある距離（メタボールの半径）以上遠ざかるとその値は0になる。逆にその中心点に近づくにつれて濃度は次第に大きくなり，あるスレッショルド（例えば1）を超えると，そのメタボールは可視化する。他のメタボールの影響がない場合，濃度が等しい点の集合は単純な球面になるが，別の複数のメタボールが近くにあると，ある点における濃度はそれぞれのメタボールの中心からその点までの距離によって定まる濃度の合計になるので，球面の表面が引っ張り合って盛り上がったり，球面と球面が滑らかにつながり合ったりして複雑なメタ曲面を構成することができる。

また，メタボールのウエイトは負の値（例えば−10など）をとることができる。この場合は，そのメタボールの中心に近いほど値は小さくなり，中心でウエイトの値と一致する。中心から遠ざかるにつれて濃度は大きくなり，半径より大きくなると濃度は0になる。これは負メタとよばれ，メタ曲面を削る場合に使われる。

メタボールのアルゴリズム

メタボールの基本的なアルゴリズムは文献1)に詳しい。その要点は以下の通りである。

①視線と各メタボールとの交点を求める。

②視線と交わるメタボールがn個とすると，視点から各交点までの距離tをt_1〜t_nとする。

③t_1〜t_nの小さい順にポインタを用いて並べたtリストを作成する。

④視点に最も近い交点に着目し，そこから後方に向かって視線上で濃度がスレッショルド（1.0）に達する点（この点の集合がメタ表面になる）を求める。

ただし，濃度を求める関数が解析的に解けないため，この関数を4つの2次式に区分して近似解を求めるという方法をとる。この各区分点と2次式を決めるのに用いられる関数(**図3**ではei( ))が，メタボールのレンダリングのスピードと精度を左右する。

図3のei( )は，今回はパーソナルコンピュータでインスタントにメタボールを表示する実験なので，精度を重視していないためメタボールの表面に段差を生じることがある。ここで得られる各メタボールの4つの区分点から視点までの距離もtリストに加え，tリスト間の隣り合う各要素間で濃度＝1になる位置，つまり視点からメタ表面までの距離を求める。この距離が求まればその点のメタ表面上の座標が計算できるので，この点に影響を与えるすべてのメタボールの影響を考慮して法線ベクトルを求め，照明のベクトルと乗算してその点の明るさを決定する。

ただし，文献1)のアルゴリズムは球形のメタボールのみを扱っている。これでは細長い形状を表現する場合など，プリミティブの数が不必要に増えてしまう。最近ではLINKSでも楕円体の濃度の広がりをもつ楕円メタを使う場合がほとんどである。そこで図3では楕円メタを表示できるようにした。そのアルゴリズムについては，リストとその注釈を参考に解読していただきたい。

プログラムについて

図3は，PC-9801上のMS-Cで書かれたプログラムである。フレーム・バッファはデジタルアーツのHyPER FRAMEを想定しているが，フレーム・バッファまわりのルーチンを置き換えればどのフレーム・バッファでもよい。これを例えばmelという名前でコンパイルする。

次に，**図4**のようなデータを，例えばdataというファイル名で用意しておく。

mel data 〈リターン〉

でレンダリングが開始する。

今回は，楕円メタをとにかく表示してみるために最小限必要なルーチンだけを紹介した。これを実用的にインプリメントするには，boundary boxなどを用いたさまざまな高速化手法，透明体，テクスチャ・マッピング，バンプ・マッピング，アンチ・エイリアシング，互いに融合しない楕円メタのグループ分けができるデータ構造，高速でしかも精度の高いei( )などを導入する必要がある。これらは読者の手近にあるレイ・トレーシング・プログラムのソースに，この楕円メタを導入する際に試みられたい。

さらに，中心の位置がずれたところにある偏心メタなどの新しいメタボールの登場によって，レイ・トレーシングの表現力が高まることを期待する。

参考文献

1) 西村，平井，河合，河田，白川，大村：「分布関数による物体モデリングと画像生成の一手法」，電子学会論文誌，Vol.J68-D, No.4, 1985. 4

2) トーヨーリンクス：「新しいCGのあり方を目指すLINKSのすべて」，PIXEL，No.50, 1986

3) 吉村浩，塚田利夫：「トーヨーリンクスにおけるCG映像技術──アルゴリズムとその技術解説」，PIXEL，No.66, 1988

▼図　3

```
/*          meta-ball ray tracing program          */

#include <stdio.h>
#include <math.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>
#include <io.h>
#include <assert.h>
#define  PI   3.14159265
#define  THD 1.0/3.0
#define  PMAX 200          /* 最大２００個のメタボールを扱う */
#define  INFIN 1000000.0   /* 無限遠 */

struct tlist {
    int mel, spl;
    float t1, t2, t0;      /* tリストの要素 */
    struct tlist *next;
           };

struct vect {
    float x, y, z;
    };

struct vect eye, ray;

float nx, ny, nz, px, pz, py;
float sqrx, sqry, sqrz;
float dg = PI/180.0;
float tmp_x, tmp_y, tmp_z, rt_x, rt_y, rt_z;

FILE *fp;
float  a[PMAX],  b[PMAX],  c[PMAX];
float cx[PMAX], cy[PMAX], cz[PMAX];
float tx[PMAX], ty[PMAX], tz[PMAX];
int   cr[PMAX], cg[PMAX], cb[PMAX];
float wgt[PMAX];
float cc[PMAX];
float la[PMAX], lb[PMAX], lc[PMAX];
float  lx, ly, lz, amb;
float daen(), sgn(), sqr(), ei(), wi();
float pix_r, pix_g, pix_b;
int jx, iy, pmax;
int b_r, b_g, b_b;
int y_b, y_e, x_b, x_e;
float r00, r01, r02, r10, r11, r12, r20, r21, r22;

main (argc, argv)
int argc;
char *argv[];

{  long ts, te;
   float  sz = 100.0;
   float sx, sy, op;
   float lcx, lcy, lcz;

float tmp_x, tmp_y, tmp_z;

   int i, j, k;

   time(&ts);
/*-------------------------------------------------------------------*/
        fp = fopen(argv[1], "r");
             if (fread (fp,  0) != 0)
                 inerror ();
/*-------------------------------------------------------------------*/
for ( k = 0; k < pmax; k++){

        a[k] = sgn(a[k])/a[k]/a[k];
        b[k] = sgn(b[k])/b[k]/b[k];
        c[k] = sgn(c[k])/c[k]/c[k];

        lcx = eye.x-cx[k]; lcy = eye.y-cy[k]; lcz = eye.z-cz[k];

/*   rotate primitive   */
rot(tx[k], ty[k], tz[k]);

tmp_x = (a[k]*r00*r00+b[k]*r10*r10+c[k]*r20*r20);
tmp_y = (a[k]*r01*r01+b[k]*r11*r11+c[k]*r21*r21);
tmp_z = (a[k]*r02*r02+b[k]*r12*r12+c[k]*r22*r22);
tx[k] = 2.0*(a[k]*r01*r02+b[k]*r11*r12+c[k]*r21*r22);
ty[k] = 2.0*(a[k]*r00*r02+b[k]*r10*r12+c[k]*r20*r22);
tz[k] = 2.0*(a[k]*r00*r01+b[k]*r10*r11+c[k]*r20*r21);

a[k] = tmp_x;
b[k] = tmp_y;
c[k] = tmp_z;
/*-------------------------*/
        cc[k] = lcx*lcx*a[k]+lcy*lcy*b[k]+lcz*lcz*c[k]-1.0;
        cc[k] = cc[k]+lcy*lcz*tx[k]+lcx*lcz*ty[k]+lcx*lcy*tz[k];
        la[k] = lcx*a[k]; lb[k] = lcy*b[k]; lc[k] = lcz*c[k];

 }
/*-------------------------------------------------------------------*/
rot(rt_x, rt_y, rt_z);
/*-------------------------------------------------------------------*/
    outp (0xee, 0xc0); /* ハイパーフレーム初期化 */
    outp (0xd0, 0xa1);
/*-------------------------------------------------------------------*/
/* main loop */
  for (i = y_b; i  <= y_e; i ++ ) { sy = 200-i; printf("Line %d¥n", i);
      for (j = x_b; j <= x_e; j ++  ) { sx = j-320;
                if ( kbhit() ) exit(0);

                jx = j; iy = i;
                  op = sqrt(sx*sx+sy*sy+sz*sz);

                  tmp_x = sx/op;
                  tmp_y = sy/op;
                  tmp_z = sz/op;

                 ray.x = tmp_x*r00+tmp_y*r01+tmp_z*r02;
                 ray.y = tmp_x*r10+tmp_y*r11+tmp_z*r12;
                 ray.z = tmp_x*r20+tmp_y*r21+tmp_z*r22;

                  if (trace()) dsp();
                  else back();
  }
 }
/*-------------------------------------------------------------------*/
   time(&te);
   printf("¥nTIME = %ld(sec)¥n", te-ts);
/*-------------------------------------------------------------------*/
}

trace()
{
static int k, kk, flg;
static float bright, td, tm, hi, da, db, dc, dd, tt, hx, hy, hz;
static struct tlist list[50];
static struct tlist *pt, *np, root;
static float ds, cmp, op, tmpa;
static float t[] = { 0.0, 0.0, 0.0, 0.0, -INFIN };
static float cos_a, phong, rfx, rfy, rfz;

               sqrx = ray.x*ray.x;
               sqry = ray.y*ray.y;
               sqrz = ray.z*ray.z;

/* 交点を求める */

           for (k = kk = 0; k < pmax; k++)
           {
                if ( ( list[kk].t1 = daen(k, &list[kk].t2) ) != INFIN  ) {
                       list[kk].t0 = list[kk].t1;
                       list[kk].mel = k;
                       list[kk].spl = 0;
                       kk++;

           }

if( kk != 0 ) {
da = db = dc = 0.0;

flg = 0;
for (;;) {

if ( flg == 0 ) flg = 1;
else { ++(pt->spl); pt->t0 = t[pt->spl]; }
/* tリストのソート */
root.t0 = -100000.0; mklist(&root, list, kk);
for (k = 0; k < kk; k++) mklist(&list[k], list, kk)
/*-----------------------------------------*/

        pt  = root.next;
        np  = pt->next;

        if ( np == NULL && pt->spl > 2 ) return 0;

        td = ((pt->t2) - (pt->t1))*0.5;
        tm = pt->t1 + td;

        hx = eye.x+tm*ray.x-cx[pt->mel];
        hy = eye.y+tm*ray.y-cy[pt->mel];
        hz = eye.z+tm*ray.z-cz[pt->mel];
```

```
/* hi は楕円メタの中心と視線を結ぶ点の中で最も濃度が高くなる点までの距離 */

         hi = sqrt(hx*hx*a[pt->mel]+hy*hy*b[pt->mel]+hz*hz*c[pt->mel]
            +tx[pt->mel]*hy*hz+ty[pt->mel]*hx*hz+tz[pt->mel]*hx*hy);

         ds = td*(1.0 - ei(hi));

        t[0] = (pt->t1);     /* t[0]～t[3] は、4つの区分点 */
        t[1] = (pt->t1) + ds;
        t[2] = (pt->t2) - ds;
        t[3] = (pt->t2);

        switch(pt->spl) {

        case 0:
                tmpa = wi(hi, pt->mel)/(ds*td);
                break;
        case 1:
                tmpa = wi(hi, pt->mel)/(ei(hi)*-ds*td);
                break;
        case 2:
                tmpa = wi(hi, pt->mel)/(ei(hi)*ds*td);
                break;
        case 3:
                tmpa = wi(hi, pt->mel)/(-ds*td);
                break;
               }

                da += tmpa;         /* デルタa, b, cの加算 */
                db += -2.0*tmpa*t[pt->spl];
                dc += tmpa*t[pt->spl]*t[pt->spl];

 if ( da == 0.0 ) continue;

        dd = db*db-4.0*da*(dc-1.0);
        if ( dd > 0.0 ) tt = (-db + sqrt(dd))*0.5/da;
         else continue;
        if (tt < 0) continue;
        if ( t[pt->spl + 1] < np->t0 ) cmp = t[pt->spl + 1 ];
        else cmp = np->t0;
        if (cmp == -INFIN) cmp = np->t0;

        if ( pt->t0 < tt && tt < cmp ) {
     px = eye.x+ray.x*tt; py = eye.y+ray.y*tt; pz = eye.z+ray.z*tt;

     getnorm(root.next);

ref(&rfx, &rfy, &rfz);
cos_a = rfx*lx+rfy*ly+rfz*lz;
if (cos_a < 0.0) cos_a = 0.0;
phong = cos_a*cos_a*cos_a*cos_a*cos_a*cos_a; /* ハイライトの大きさ */

        bright = nx*lx+ny*ly+nz*lz+phong+amb;
        if (bright < 0.0) bright = 0.0;

        pix_r = pix_r*bright;
        pix_g = pix_g*bright;
        pix_b = pix_b*bright;

           return 1;
                       }

    } /* end of loop */
   }
else return 0;
 } /* end of trace() */

float daen(k, l) /* 楕円の交点を求める */
int k;
float *l;
{
   float aa, bb, d;
   float lcx, lcy, lcz;

   lcx = eye.x-cx[k]; lcy = eye.y-cy[k]; lcz = eye.z-cz[k];
   aa = sqrx*a[k]+sqry*b[k]+sqrz*c[k];
   if ( aa == 0.0 ) return INFIN;
   bb = 2.0*(ray.x*la[k]+ray.y*lb[k]+ray.z*lc[k]);
/* 回転 */
   aa = aa+ray.y*ray.z*tx[k]+ray.x*ray.z*ty[k]+ray.x*ray.y*tz[k];
   bb = bb+(lcz*ray.y+lcy*ray.z)*tx[k]+(lcx*ray.z+lcz*ray.x)*ty[k]+(lcx*ray.y+lcy*ray.x)*tz[k];
/* ---------------------------------- */
   d = bb*bb-4.0*aa*cc[k];
   if (d < 0.0)  return INFIN;
   *l = (-bb + sqrt(d))*0.5/aa;
   return ( -bb - sqrt(d))*0.5/aa;
}
```

```
dsp() /* メタボールの色を表示する */
 {
 int rr, gg, bb;

 rr = (int)pix_r;
 gg = (int)pix_g;
 bb = (int)pix_b;

 if (rr > 255) rr = 255;
 if (gg > 255) gg = 255;
 if (bb > 255) bb = 255;

  cgpset(jx, iy, rr, gg, bb);

 }

 back() /* 背景色を表示する */
 {
  cgpset(jx, iy, b_r, b_g, b_b);
 }

 float sgn(v) /* 符号を求める */
 float  v;
 {
  if (v != 0.0) return (( v < 0.0 ) ? -1.0 : 1.0);
  else  return 0;
 }

 skip (f) /* スペース文字をスキップする */
 FILE   *f;
 {
     int ch;

     while (isspace (ch = getc (f)) && ch != EOF);
     return ch;
 }

 fread (f, chr) /* データーの読み込み */
 FILE   *f;
 int     chr;
 {
     int     ch, i, j,n;
     FILE   *f2;

     pmax = 0;

     while (1) {
         switch (ch = skip (f)) {
             case ';':
                         while ((ch = getc (f)) != '¥n' && ch != EOF);
                         break;
             case '{':
                         if (fread (f, chr) != 1)
                         inerror ();
                         break;
             case '}':
                         return 1;
             case 'p':

                         fscanf(f, "%f%f%f", &a[pmax], &b[pmax], &c[pmax]);
                         fscanf(f, "%f%f%f", &cx[pmax], &cy[pmax], &cz[pmax]);
                         fscanf(f, "%f%f%f", &tx[pmax], &ty[pmax], &tz[pmax]);
                         fscanf(f, "%d%d%d", &cr[pmax], &cg[pmax], &cb[pmax]);
                         fscanf(f, "%f", &wgt[pmax]);
                         ++pmax;
                         break;
             case 'c':
                         fscanf(fp, "%f%f%f", &eye.x, &eye.y, &eye.z);
                         fscanf(fp, "%f%f%f", &rt_x, &rt_y, &rt_z);
                         fscanf(fp, "%f%f%f", &lx, &ly, &lz);
                         fscanf(fp, "%f", &amb);
                         fscanf(fp, "%d%d%d", &b_r, &b_g, &b_b);
                         fscanf(fp, "%d%d%d%d", &y_b, &y_e, &x_b, &x_e);
                         break;
             case EOF:
                         return 0;
             default:
                         inerror ();
         }
     }
 }

 inerror () {
     fprintf (stderr, "Input error¥n" );
     exit(0);
 }
```

```
mklist(p, q, r) /* tリストを作る */
struct tlist *p, *q;
int r;
{
 int k;
 float dt;
 float lst = INFIN;
 for (k = 0; k < r; k++) {
     if ( (q + k) != p ) {
      dt = (( q + k )->t0 ) - ( p->t0 );

  if (( dt < lst) && ( dt >= 0.0 ) && !( dt == 0.0 && q + k < p ))
                                  { lst = dt; ( p->next ) = q + k; }

                      }
                      }
        if ( lst == INFIN ) ( p->next ) = NULL;
}

float wi(di, i) /* メタボールの分布関数 */
float di;
int i;
{

      if (( 0.0 <= di ) && ( di <=  THD ))
                      return (wgt[i]*(1.0-3.0*di*di));
      else if (( di >=  THD ) && ( di <= 1.0 ))
                      return (1.5*wgt[i]*(1.0-di)*(1.0-di));
           else return 0.0;
}

float ei(r) /* メタボールの近似計算のための関数 */
float r;
{
       if (r < 0.22) return (-0.5*r+0.33);
       return (0.346*r+0.243846);

}

getnorm(p)      /* 放線ベクトルの計算 */
struct tlist *p;
{
 float tpx, tpy, tpz;
 float r, pw, nxs, nys, nzs, op;
 int k;
   pix_r = pix_g = pix_b = 0.0;
   nxs = nys = nzs = 0.0;
 do {
   k = p->mel;
tpx = px-cx[k]; tpy = py-cy[k]; tpz = pz-cz[k];
r = tpx*tpx*a[k]+tpy*tpy*b[k]+tpz*tpz*c[k]+tx[k]*tpy*tpz+ty[k]*tpx*tpz+tz[k]*tpx*tpy;
if (r < 0 ) r = 0;
   pw = wi(sqrt(r), k);

          nxs += (a[k]*tpx+0.5*(tz[k]*tpy+ty[k]*tpz))*pw;
          nys += (b[k]*tpy+0.5*(tz[k]*tpx+tx[k]*tpz))*pw;
          nzs += (c[k]*tpz+0.5*(ty[k]*tpx+tx[k]*tpy))*pw;

          pix_r += (float)cr[k]*pw;
          pix_g += (float)cg[k]*pw;
          pix_b += (float)cb[k]*pw;

    } while ( ( p = p->next ) != NULL );

          nx = nxs;
          ny = nys;
          nz = nzs;

          op = sqrt(nx*nx+ny*ny+nz*nz);
          if (op == 0.0) op = 1.0;

          nx = nx/op;
          ny = ny/op;
          nz = nz/op;

}

cgpset(v, w, r, g, b) /* ハイパーフレームの1ドット表示ルーチン */
int v, w;
char r, g, b;
{

   outp(0xd4, v );
   outp(0xd5, v >> 8);

   outp(0xd6, w);
   outp(0xd7, w >> 8);

   outp(0xd8, r);
   outp(0xda, g);
   outp(0xdc, b);
}

ref(rx, ry, rz) /* 反射のベクトル計算 */
float *rx, *ry, *rz;
{
float w;
        w = -2*(ray.x*nx+ray.y*ny+ray.z*nz);
        *rx = ray.x+nx*w;
        *ry = ray.y+ny*w;
        *rz = ray.z+nz*w;
}

rot(x, y, z) /* 回転 */
float x, y, z;
{
float snx, cnx, sny, cny, snz, cnz;

cnx = cos(x*dg); snx = sin(x*dg);
cny = cos(y*dg); sny = sin(y*dg);
cnz = cos(z*dg); snz = sin(z*dg);

r00 = cny*cnz; r01 = snx*sny*cnz-cnx*snz; r02 = cnx*sny*cnz+snx*snz;
r10 = cny*snz; r11 = snx*sny*snz+cnx*cnz; r12 = cnx*sny*snz-snx*cnz;
r20 = -sny   ; r21 = snx*cny            ; r22 = cnx*cny;

}
```

▼図　4

```
{
{
p
78.000000 63.000000 63.000000        楕円の3軸の大きさ
0.000000 70.000000 0.000000          楕円メタの位置 (x, y, z)
0.000000 0.000000 -90.000003         楕円メタの回転角度 (x, y, z)
100 200 250                          楕円メタの色 (r, g, b)
1.500000                             重み

p
31.500000 12.369316 12.369316
0.000000 39.500000 0.000000
0.000000 0.000000 -90.000003
200 130 100
1.500000

p
67.980881 77.283524 77.283524
-1.000000 -14.500000 0.000000
0.000000 0.000000 -90.000003
200 220 60
7.344886

p
117.613770 28.301943 28.301943
-74.000000 33.000000 0.000000
0.000000 0.000000 37.746805
20 200 100
1.500000

p
123.547562 26.832817 26.832817
77.000000 33.000000 0.000000
0.000000 0.000000 -29.054604
20 100 200
1.500000

p
16.020582 14.504354 24.657402
-1.000000 -19.000000 -49.000000
0.000000 0.000000 -95.194430
20 20 200
6.912585
}

c
0 0 -150.000000          カメラの座標 (x, y, z)
0 0 0                    カメラの回転角 (x, y, z)
0 0 -1                   照明の単位ベクトル
.5                       アンビエント
210 120 180              背景の色
100 300 220 420          表示スクリーンの領域 (y0 y1 x0 x1)

}
```

3次元設計用CAD機論

設計を支援する3次元CADの機能と，それを支えるソフトウエアとハードウエアの要素技術について述べる。

田中　秀樹*
山本　理**
木下　篤***

はじめに

先日，ある会社のCAD導入検討を担当している方と会う機会があり，その中で彼は次のようなことを話してくれた。

「社内の技術部門は大きく分けて，デザイン部門，設計部門，生産部門の3つの部門からなっており，導入しようとするCADシステムに対する要求も当然，それぞれの部門で全く異なる。まず，デザイン部門は，商品のデザイン検討やプレゼンテーションなどのために，システムのコンピュータ・グラフィックス（CG）機能に興味がある。生産部門はもちろん，CAM，NCデータの出力について強力なシステムを欲しがっている。ところが設計部門はというと，どうもCADシステムを使いたがらない。」

この会社の設計部門では，元来図面を書くという作業が全体の1割程度でしかなく，製図のためのCADを導入するメリットは少ないということである。設計の中の製図以外の作業は，例えば，部品の重量計算や，簡単な機構解析や干渉チェック，規格とのマッチング，動作シミュレーション，構造解析などと思われる。

おそらくこの会社の設計部門の方々は，自分たちが行っているさまざまな設計作業を，コンピュータ上で行えるはずがないと思っているのであろう。これらの作業はそれぞれ複雑で互いに絡み合い，いわゆる設計の試行錯誤とよばれる作業である。

われわれ，システムを開発する者の立場から言わせてもらえば，これらのどの作業要素も不可能なものはない。しかし，これらの作業のすべてが，「統合化されているか」「設計者にとって使いやすいか」と問われると，現在のほとんどの3次元CADは決して満足のいく解答を与えていないことを認めざるを得ない。

当社は総合システム・サプライヤとして，CAD/CAMを中心とした総合システムの販売とその技術サポートを商品として提供している。2次元設計製図用CADシステムとして，エンジニアリング・ワークステーション（EWS）ベースのGRADE/DPがあり，3次元汎用設計CADとしてGRADE/Gがある。また，メカニカル分野のアプリケーション・ソフトとして，GRADE/GをベースにCAM，金型設計，配管設計（**図1**），機構解析，構造解析などを扱ったシステムがある。特に，複合自由曲面切削・3次元測定機データ処理・DNC機能を有し，金型設計分野では評価・実績を得ている。

これらに加え，数年前より機械設計を中心とした設計全般の支援ツールとしての3次元CADを，ソリッド・モデリングの重要な応用分野として注目し開発を行っている。

本稿では，われわれが目指す3次元設計用CADの機能と，これを構成するのに不可欠な技術要素について，システム提供者の目から整理を試みる。まず，設計を支援するシステムの機能について述べ，次に，このシステムに必要な要素技術についてまとめる。

設計を支援するシステム

常に設計者の傍らにあり，設計の道具として設計者に使ってもらうために，システムにはいろいろな機能が要求されている。CAE（Computer Aided Engineering）という言葉がこれらの機能の総称であるといえるであろう。

*たなか　ひでき　　日立造船情報システム㈱　営業統括室　〒108　東京都港区高輪1-4-10　　**やまもと　おさむ　　同社開発本部第1開発部
***きのした　あつし　　同社CAEセンター

ここでは，これを以下の３項目に区分してまとめる。

- ●形状解析
- ●ビジュアル・シミュレーション
- ●構造解析

■形状解析

設計者は，最初は曖昧な設計イメージから，さまざまな設計要件や製品の機能仕様を満足させながら徐々に形状を規定していく。このような設計の試行錯誤の過程に対して，システムは以下のような形状解析機能を提供できる。

- ●任意断面形状の生成
- ●断面特性の計算
- ●立体のマスプロパティ計算
- ●材料物性データや物理公式などの登録・参照
- ●２物体の干渉形状の算出
- ●２物体間の最短距離点の算出
- ●曲面部分の曲率計算，および最大曲率点の算出

例えば，任意断面形状の生成は，単に図面に断面図を付け加えるためだけに必要なのではない。作成中の３次元モデルが設計要件に適合しているか（例えば，肉厚や端部の最小Ｒなど）をチェックするためにも使用される（**図２**）。また，ある断面での断面積・断面２次モーメント・断面係数などの断面特性を計算させ，この数値を使用してより細部の形状を決定していくということも可能である。

立体のマスプロパティ計算機能により求まる体積や表面積を利用して，その製品を作るのにかかる材料や塗料の見積りをしたいという要求は多い。さらに，組み立てられた製品全体の重心位置や慣性モーメントを求めることができるので，製品のバランスやパッキングのシミュレーションなども可能である。

鋳鉄やアルミニウムなど，主な物質の密度・ヤング率・ポアソン比などの物理特性をデータベース化してシステム内にもち，かつ一般的な物理公式や頻繁に使う計算式を登録できる機

◉図２は次ページ。

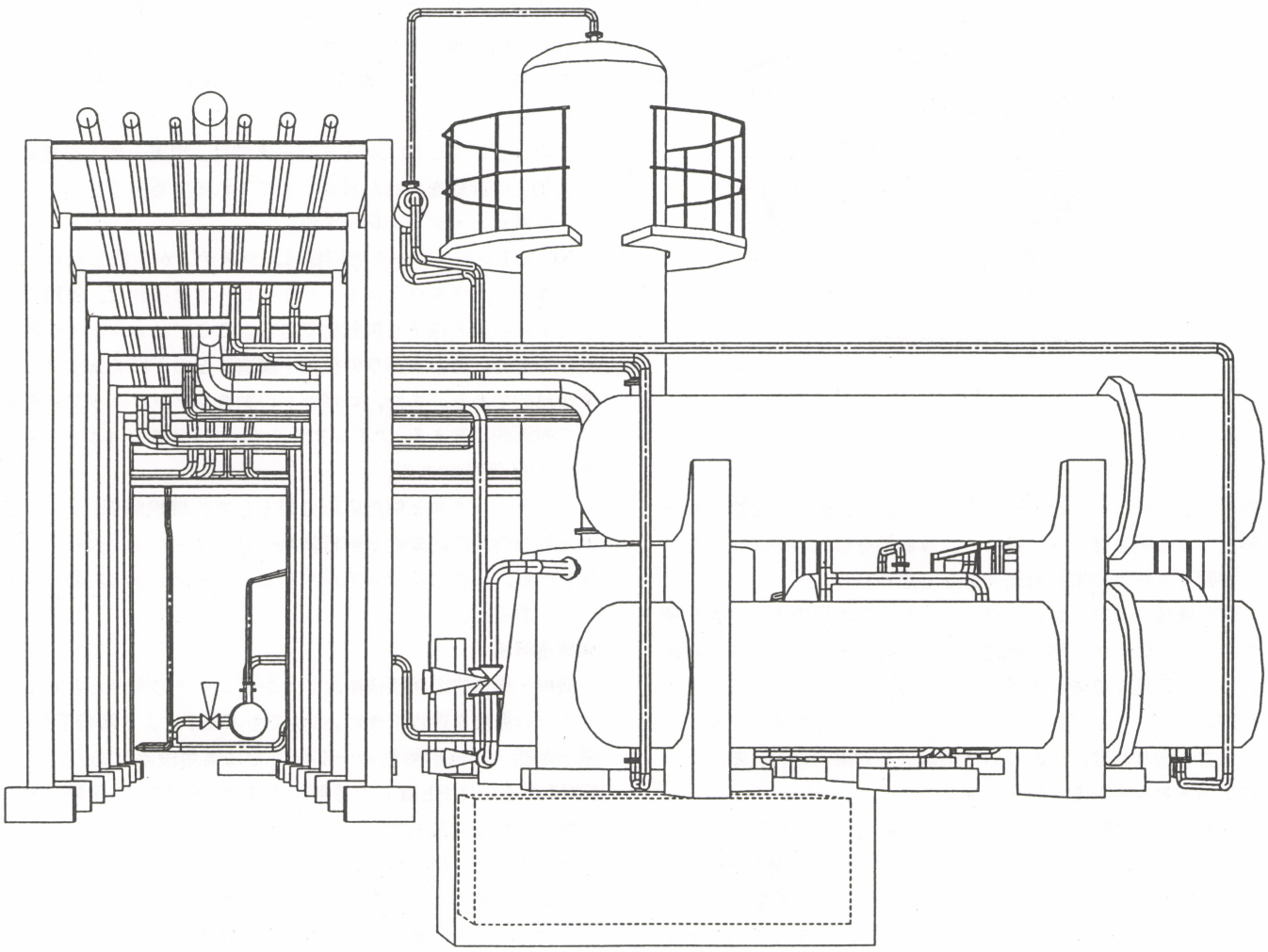


図１　隠線消去された配管組図

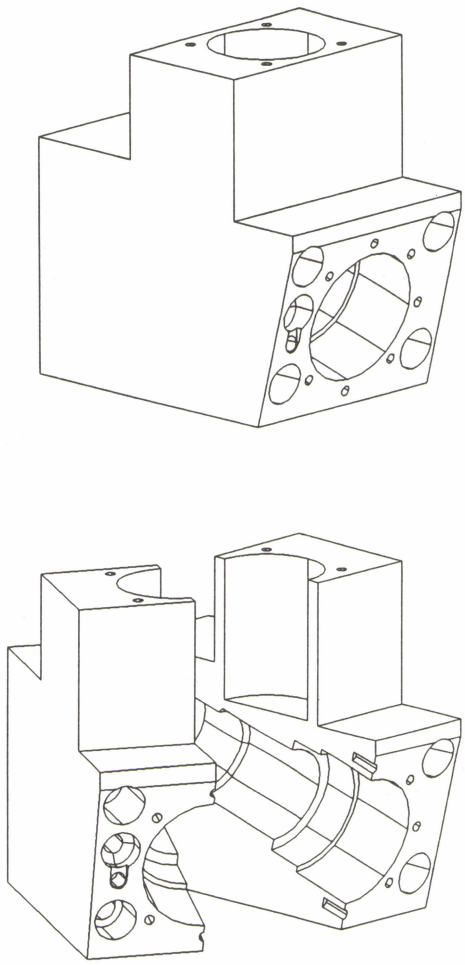


図2　機械部品の切断面

能が用意されていれば便利である。公式や計算式の変数の値を形状から参照させたり，逆に，計算結果を形状の修正コマンドが参照できれば非常に有効であると思われる。

精密機械の設計においては，部品どうしの干渉チェックや最短距離点の算出は必須の機能である。限られたスペースの内側に多くの部品を最適に配置したり，機構部の当たりチェックを正確に行うことが可能である。これに加え，3次元グラフィックスの力を借りれば，これらの作業はより効果的に行える。

■ビジュアル・シミュレーション

近年のCGに関するハードウエアの発展は目覚ましく，さまざまな分野のコンピュータ・システムの可能性を増大させている。今後10年間のコンピュータ技術の中で最も重要なテーマは，AIやニューロコンピュータなどではなく，CGであるとさえ言う人もいる。

設計用CADの周辺にあるビジュアル・シミュレーションの例を取り上げてみよう。

1)　デザイン・シミュレーション

設計者の頭の中にあるデザインイメージは，設計者自身でも明確化することが難しく，また他人に伝えにくい。ここで，設計者の頭の中のそのデザインイメージに，設計者と対話しながらどこまでリアルに迫れるかがデザイン・シミュレーションの意味である。このことによって，設計者はデザインの意図をより洗練させたかたちで製品形状に盛り込むことができる。

カラーリング・質感の表現・半透明表示・光源の設定・ライティング計算などは，現状ではすべてハードウエア化されており，ダイナミックなシミュレーションが可能である。

機械設計用CADという意味合いからいえば，過度のCG機能は不要であるが，上記程度の機能は3次元対話システムの標準機能になりつつあるといえる。

2)　NC加工シミュレーション

システムが計算したNC加工データを実際のNC工作機にかけるのと同じイメージで，NC加工シミュレータはディスプレイ上に表示してくれる。工具の実際の動きをアニメーション化してそれを自由なアングルから見回すだけでなく，手元のダイアルを使って逆再生や正再生を何度も見ることが技術的には可能であるし，工具と切削対象物との干渉部分を強調表示したり半透明表示することもできる。また，工具の動きに合わせて工具と切削対象物との干渉部分を除去していけば，カッターマークの残った加工形状が現れる様子も視覚化できるはずである。

このようにNC加工シミュレータでは，現実の加工を見る以上に加工の詳細をわれわれに示してくれるであろう。

3)　ロボット・シミュレータ他

NC加工シミュレータと同様に，システムが生成したロボットの制御データをロボット・シミュレータに送れば，実際のロボットの動作の様子を視覚化できる。ここではロボットや部品だけでなく，そのまわりの機械や構造体との干渉チェックも重要な目的である。また，ロボットの動きをグラフィックスを通じて対話的に教える，“インタラクティブ・オフライン・ティーチング”も可能である。

ロボティックスと似た応用分野としての機構解析も，その3次元的な対話の必要性と解析結果のアニメーション表示という性質から，まさにビジュアル・シミュレーションの対象分野の一つであろう。

■構造解析

機械・構造物の解析技術は，コンピュータの発達とともに目覚ましい発展を遂げてきた。なかでもFEMによる解析手法は，線形解析・非線形解析の一部がすでに実用段階にあり，またFDMあるいはBEMによる解析手法は，流体力学・伝熱工学の分野で重要な役割を果たしている。

このように一口に解析といっても種々の手法があり，また目的に応じてさまざまな数理モデルを設定して行われる。

一般に製品の開発・設計にあたり，解析技術を次のように分類することができる。

(1)　製品の応力・変形の評価を行い，信頼性向上を目的とした強度信頼性解析（応力解析，振動解析）

(2)　製品の機能・性能向上を目的とした性能解析（流体解析，

熱伝導解析)

(3) 製品製作時における成形・加工解析（樹脂流動解析，凝固解析)

(1)で使用される解析ソフトウエアは，製品の種類によらない汎用性のある解析プログラム（NASTRAN，SAP，MARC，ADINA)が通常よく用いられる。(2)，(3)では汎用の解析プログラムだけでは不十分な場合が多く，過去の実績や試験データを用いたり，設計ノウハウを盛り込んだ専用の解析プログラムを使用することが多い。

このように解析技術では各種解析プログラムを利用して製品の信頼性・性能を評価・検討するわけであるが，それらの多くは相反する結果を生み出すことが多く，そのためバランスのとれた解析技術が要求される。

さて近年では，コンピュータ内部に設計対象の形状をモデリングし，各種数理モデルと数値解析手法によって製品の機能・性能を評価するシステム(CAE)が提案されつつある。これは，開発・設計段階において従来の試作→試験→修正→試作というループをコンピュータ内部で実現することによって，開発・設計期間の短縮，コスト低減を可能にするものである。

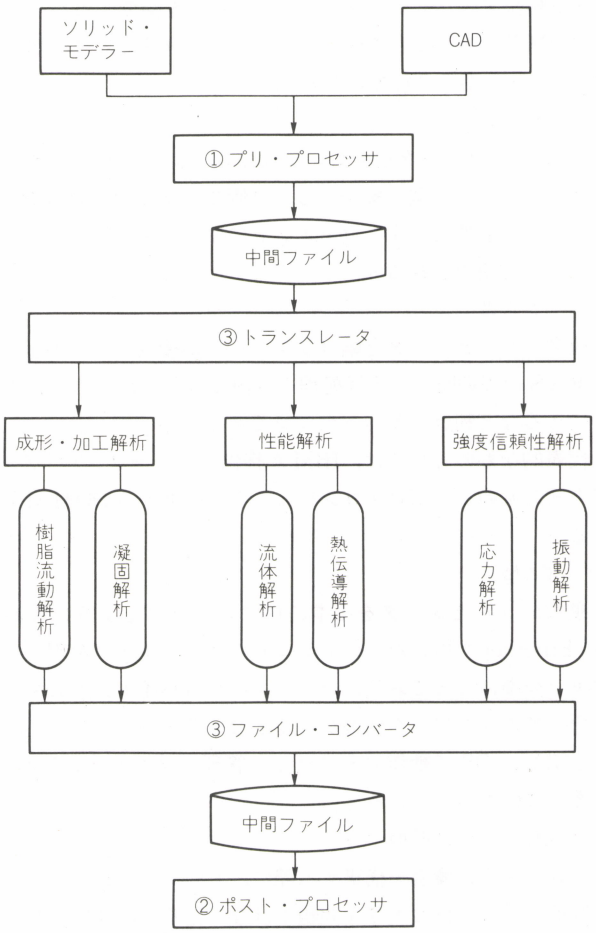


図3　解析システムの流れ

解析システムの一例を**図3**に示す。システムを構成する各ソフトウエアについて簡単に説明する。

1)　プリ・プロセッサ

本来解析データの作成には膨大なマンパワーを要するだけでなく，人為的ミスなどによって解析結果の信頼性を損ねることもある。プリ・プロセッサはこれらの問題を解決する有力な解析支援ツールである。すなわち，1次元，2次元，3次元の各有限要素を自動的に生成する機能を有し，これら要素をグラフィック上に表示することによって容易にモデルチェックを行うことができる。通常，CADによってすでに幾何形状モデルが作成されている場合が多く，これをベースに有限要素にモデリングできれば作業効率も上がる。

2)　ポスト・プロセッサ

解析の結果は製品の開発・設計に重要な情報を与える。ポスト・プロセッサはCG機能によってさまざまな情報をビジュアルにかつスピーディーに表示するだけでなく，必要とする情報の編集，リスティング，さらには各種ルールによる評価チェック機能を備えたものもある。

3)　トランスレータ／ファイル・コンバータ

解析プログラムの多くは固有の入力／出力フォーマットをもち，一方，プリ／ポスト・プロセッサは共通ファイルを介してデータの受け渡しが行われる。したがって，各構成プログラムがフレキシブルに結合されたシステムを構築するには，トランスレータとファイル・コンバータの役割が重要となる。トランスレータは，プリ・プロセッサが作成する共通ファイルより各種解析プログラムの入力フォーマットに応じてデータを作成する。ファイル・コンバータは，解析プログラムの出力データをポスト・プロセッサの入力ファイルにより作成する。

設計において重要な要素技術の一つである解析技術は，最近の各種CAEシステムの提案によって，設計者にとってかなり身近なものとなっている。しかし，優れた解析技術や解析システムが提案されても，これを利用する解析者の能力や経験が非常に重要である。例えば，モデル化によってコンピュータに負荷をかけずかつ精度の良い結果を得るには，解析者のセンスによることが多いのも現状である。

図4～6にサーフィス・モデルをシェーディングした例，**図7**に解析例を示す。

システムに必要な要素技術

設計を支援するシステムは以前からいろいろな場で論じられてきたが，ユーザーからはまだまだ現実に使用できるシステムとみなされ難かったように思う。これはそれぞれのシステムの基本となる技術が未成熟だったことに起因していると考えられる。近年，それぞれの技術はある程度のレベルに達してきて，実用面からみても問題のない設計支援システムが数多く発表されている。ここで，システムを構成するそれら要素技術について簡単ながら解説を試みる。

以下で述べるのは，特に重要と判断される次の要素技術に限定する。

●ワークステーション
●ネットワーク
●マン・マシン・インタフェース
●形状表現

■ワークステーション

ワークステーションを，価格と機能を合わせた面からハイエンドとローエンドに便宜上分類して話を進める。

1)　ハイエンド

グラフィック用ハードウエアは，GDT（Graphic Display Terminal），GWS（Graphic Work Station）となって高速，高機能化している。

GDTは表示機能だけを高速・高機能に行う専用マシンで，表示の制御自体は他のホスト・コンピュータを使用するためホストとの通信ネックやコストが高くなるなどが考えられるが，機能・性能ともに十分満足できるものがでてきている。

GWSは表示機能を高めたワークステーションであり，単体で表示の制御まで行い，現在，各種適用分野では主流を占めており，グラフィック性能もGDTに迫りつつある。

例として，現在われわれが提供する最上位システムを紹介する。それはEWSまたはスーパー・ミニコンにGDTを使用したGRADE/NC-CUBEで，3次元のモデリング，マシニング・センターなどのNCデータ作成機能を有する。そこでは3次元形状の作成を行い，任意の時点で視点を変えながら形状を確認でき，また光源の設定や色の濃淡などをダイナミックに変化させながら表現できる。システムで使用しているGDTでは，以下のグラフィック表示機能・性能を有している。

●機能
・3次元図形をダイナミックに回転・拡大・縮小表示
・最大8光源までのライティング機能
・グロー（Gouraud）およびフォン（Phong）シェーディング
・Ethernetを用いた高速インタフェース
●性能
・42万ショート・ベクトル／秒
・44,000ポリゴン／秒
・1,677万色同時表示
・1,280×1,024の画面解像度

われわれのシステムは前節で述べたビジュアル・シミュレーションに対応するものと考えており，このようなハードウエアを持って初めて実用に耐え得る支援システムを提供できるにいたった。

2)　ローエンド

今までのハイエンド/ローエンド・マシンの区別は正確ではなくなってきている。ワークステーションの発達はわれわれシステム開発者の予測をはるかに上回るものがある。2，3年前までは個人での使用を前提としたローエンド・マシンの処理能力の今後もつべき最良の値として，以下の3個の“1M”を想定していた。すなわち，

●画面解像度…1,024×1,024＝1Mピクセル
●主記憶…1Mバイト
●処理速度…1MIPS

である。それが現在では画面解像度はその目標値にほぼ一致したが，メモリは8Mバイト，処理速度は4MIPSになろうとしている。

このような変化が現れたのは1980年ころからであり，EWSが高機能を維持しながら価格を下げ，パーソナルコンピュータが低価格を維持しながら機能を上げてきた。3次元形状をリアルに表現し，ダイナミックにその表示を変化させていくような負荷の高い処理をハイエンド・マシンで行い，文書作成や部品図の作図・出図をするような処理はローエンド・マシンで行う機能分散型の作業環境になってきている。

■ネットワーク

ハイエンドとローエンドのマシンを有機的に結合させ，機能分散処理形態を実現するために不可欠な技術要素としてネットワークがある。ネットワークはEWSとともに発達し，そのどちらかの成長率が低くても今日の状態は実現しなかったと考えられる。

ここで，ネットワークを利用した設計システム環境を考えてみる。

1)　ネットワーク・トポロジ

ネットワークを利用してEWSを接続する形態（トポロジ）について解説する。トポロジはネットワークを構成するハードウエアや制御ソフトにも依存するが，大きく分けてスター型，リング型，バス型の3種類になる。

(1)　スター型

1つのコンピュータを中心に複数のローエンド・コンピュータを星（スター）型に接続する。高価で複雑な制御装置やソフトウエアを必要とせず，ローエンドのマシン用に簡易ネットワークとして使用されるのみとみられる。

(2)　リング型

複数のコンピュータを輪（リング）状に接続する。大量台数のEWSの接続時にも高速処理を維持できたり，障害時にも強くできるなど利点の多いトポロジである。

代表的なものとしては，IBMが提供するトークン・パッシング方式のものやアポロ・コンピュータが提供するDOMAINがある。

(3)　バス型

複数のコンピュータを1本の線にぶら下げるように接続する。Ethernetを用いたこのトポロジのネットワークが現在では事実上標準になっており，ローエンドからハイエンドのマシンを含んでネットワーク構築が可能となっている。

UNIXのOSで動作するファイル管理ソフトのNFSなどが標準的に使用されている。

2)　設計システム環境

設計システム環境を構築する際に，どのトポロジをユーザーが選択するかは特に大きな問題ではないだろう。問題となるのはユーザーの設計システムを正確に把握・分析し，どの機能をどのレベルのマシンに分担させるかを見極め，それに最適なネットワークを構築することである。それを実現させるだけのハードウエアの技術的基盤は整備され始めている。

設計システムの機能分類とハードウエアの関係の整理をわれ

われなりに行ってみる。

a)　外見の形状などを決める意匠デザイン→ハイエンド

b)　形状の力学的な解析とその妥当性を判断するシミュレーション→ハイエンド

c)　既存設計データを有効利用するためのデータベース処理→ハイエンドの専用マシン

d)　作図および出図→ローエンドとハイエンド

e)　文書作成→ローエンド

a)機能は3次元グラフィックスの高速・高機能表示が不可欠であり，b)機能は複雑な数値計算を，c)機能も大量のデータを高速に処理する必要がある。d)機能は全体構想図のように複雑なA1サイズの図面からA4の部品図のように簡易な処理まで考えられるからである。ハードウエアの台数まで考えるとa)～c)機能は少数，d)，e)機能は大量台数になる。

これらのハードウエアは単一のトポロジのネットワークで接続する必要はない。すなわち，おのおのの処理を行うハードウエアに見合ったネットワークを選択することになる。しかし，ネットワーク間を1つに結合する全体的なネットワークが必須であり，これをバックボーン・ネットワークと称し，高速の光ファイバを用いたものが使われている。

図8で示すようなネットワーク環境はすでに現実のものとなっており，今後さらに一般化していくであろう。

■マン・マシン・インタフェース

一般的にコンピュータ・システムと人間が対話を行う接点の処理のことをマン・マシン・インタフェースとよび，その出来具合によって，システムの機能がいくら優れていてもその価値が認められないほど重要なものと認識され始めている。

そこで先駆的な役割を果たしたシステムを概観し，今後技術的にその背後にある思想的に求められる内容について考えてみたい。

1)　先駆的なインタフェース

1980年以降のワークステーションにおけるマン・マシン・インタフェースのあるべき姿を方向付けることにMacintoshが果たした役割は無視できない。ほとんどすべての操作を画面上の機能を意味する小さな絵（アイコン）を指示してマウスで行え，複数の作業を重なって表示される窓（ウィンド）上で一度に（人間からみて）行える，などである。これらのことができるマシンは以前にも存在したが，低価格の量産品としてこれら

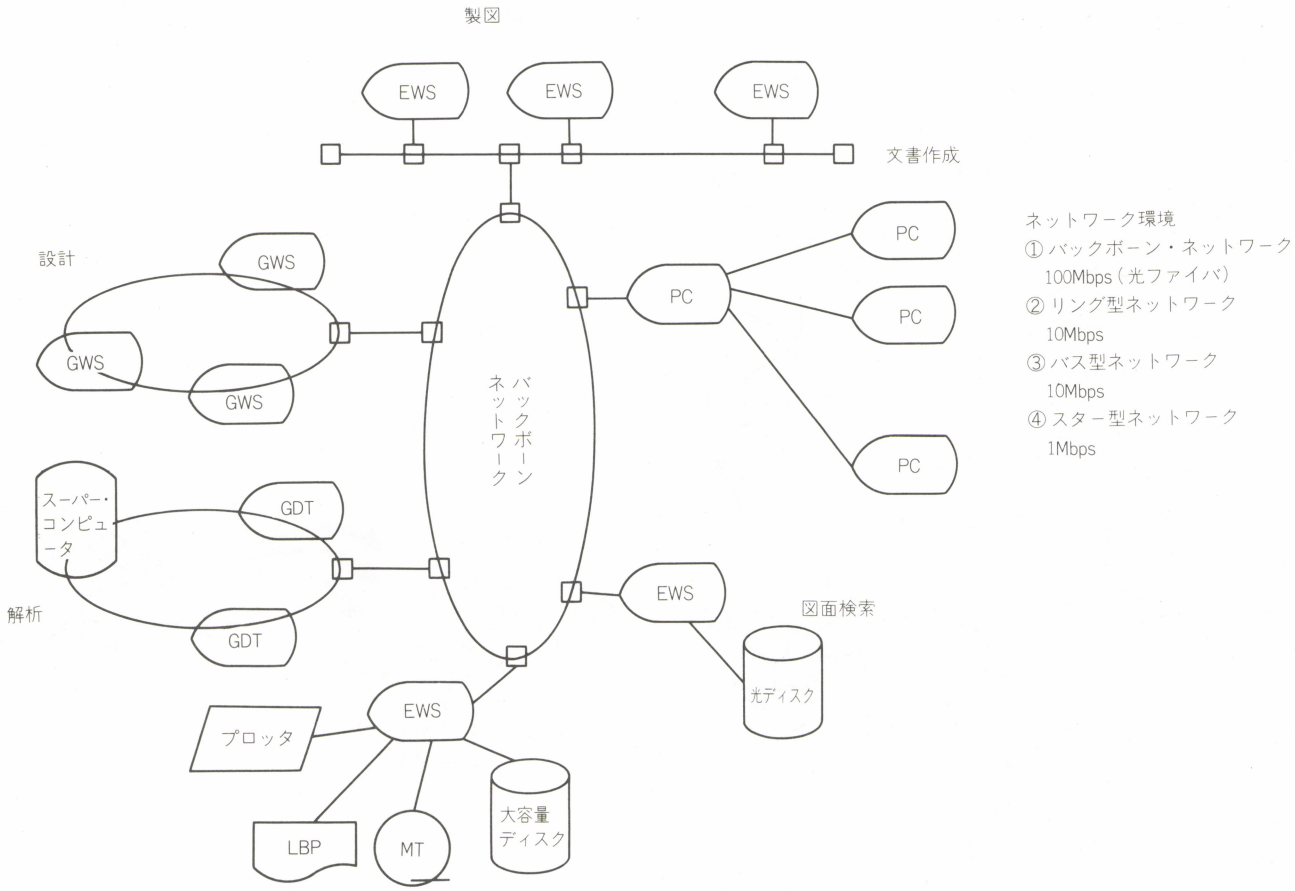


図8　設計システム環境

先駆的なインタフェースを全世界のユーザーに普及させたことがおおいに特筆されるのである。

その特徴点を整理すると次のようになる。

a) デスクトップの概念の導入
b) マルチウィンドの実現
c) プルダウン・メニューの採用
d) アイコンの採用
e) マルチジョブの実現（疑似的）

総括するとそのマン・マシン・インタフェースの良さは，操作するときにシステムと対話をする規則が視覚的に理解しやすくかつ曖昧に指示でき，操作者の思考過程に即応して複数の処理を断片的に実行できる柔軟な構造になっていた，ということであろう。

2) 設計者の机

設計者の机のまわりには何があるのだろうか。そのすべてが設計システムの中に含まれている必要がある。

- ●製図道具
- ●筆記用具
- ●標準部品一覧
- ●各種事典
- ●電卓

これらは前項までに述べた機能分散とするほどのものではなく，どの機能の中にも含まれてしかるべきものである。

例えば，ある機構の設計を行っているとき全く別の機構の設計アイデアを思いつくことは当然のようにあるだろう。そのとき，なんの苦もなく即座に覚書として残しておき，また再びもとの作業に戻れる。このようなことがすべての設計支援システムに標準的に含まれなければならない。

設計者が，気ままでわがままな人間的思考過程を中断されることなく作業を続けられることが重要である。

■形状表現

ここでいう形状表現とは，設計対象物をモデル化してコンピュータの中に表現する手法のことである。

機械部品の多くがその形状によって機能を果たしていることや，形状が解析・シミュレーション・加工・組立てなど，設計・生産にかかわるすべての情報のもとになっているという意味において，形状表現は3次元設計CADの基幹となる要素技術であり，かつ最も困難なテーマでもある。

1) 形状表現の分類

通常，形状表現方法についての議論はサーフィス・モデルとソリッド・モデルとに分けて行われることが多いが，これらの特徴や関係についてはさておき，ここでは“自由曲面を含んだソリッド・モデル”を目的の形状表現として話を進める。

内部表現として以下の分類が一般的である（**図9**）。

(1) CSG（Constructive Solid Geometry）

基本立体あるいは基本曲面の集合により物体を表現し，その作成過程をデータとしてもつ。各プリミティブとその間のオペレータ，すなわち和・差・積が主なデータとなる。B-Repや

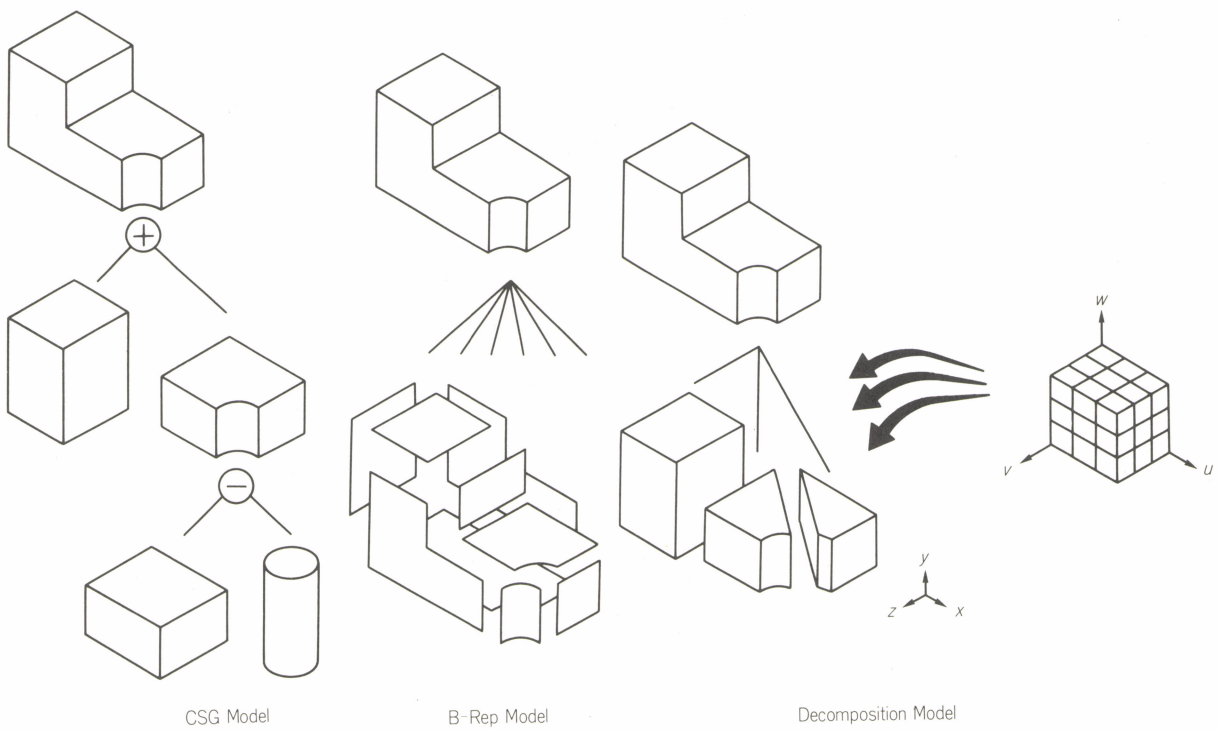


図9 主な形状表現手法

Decomposition Model などのデータへ変換が可能である。

(2) B-Rep (Boundary Representation)

物体を，これを形成する境界の情報により表現する。稜線，頂点，面などの接続関係を表す位相データをもつ。複雑なデータ構造を形状操作のたびに更新する必要があるが，境界をもった面（トリムされた面）情報を明示的にもっていることは他の表現方法にはない利点である。

(3) Decomposition Model

物体をあらかじめ定めた単純な立体ユニットの集合で表現する。CSG や B-Rep では，物体をその境界の情報からとらえようとするのに対し，この表現では物体の内空間をデータ構造として表現しようという発想が大きく異なるといえる。基本的な演算が単純で統一的であるが，一般に形状表現は近似的でデータ量が多くなる。構造解析用のデータと親密性があり，この分野への適用例が多いと思われる。

2) 表現方法の応用

現在のわれわれのモデラーでは CSG 情報と B-Rep 情報を併せ持ち，操作に応じて使い分ける方法をとっている。

CSG 情報は，その形状の成立ちや各部の特徴を表しており，内部演算やデータの応用の段階において属性データとしての意味をもつ。また，履歴操作やパラメトリックな形状修正にも利用できる。

一方，B-Rep 情報は，ワイヤー・フレーム表示やシェーディング表示のための表示データを直接出力できる。また，一意的に形状を表現しているので，体積や表面積なども高速に求めることができる。CAM や解析などの応用へ出力すべきデータもこれに近いかたちをしている。

このように，CSG 情報と B-Rep 情報は互いに相対する特徴をもっており，2 つの表現を併せ持つ考え方はもはや一般的であるといえる。しかしこのことは，データ量・計算量の増大という点でシステムに負荷をかけている。実用への適用性とのトレードオフではあるが，現実の要求はさらに高度であるため，この方向は当面変わらないものと思われる。最近の GWS の急速な進化が問題の見方を変えてくれるかもしれない。

3) 形状入力手法

形状表現に関係するわれわれの課題として，形状入力の手法の改善について取り組まなければならないと考えている。

ソリッド・モデラーの利用者から多く指摘されることであるが，現状の形状入力のプロセスは設計者の形状のとらえ方や組立て方にマッチしていない。具体的な点を以下にあげる。

- 集合演算という考え方が，現実の世界の“物作り”の仕組みとなじまない。
- 3 次元空間に対する操作概念が理解しにくい。
- 設計者の頭の中の曖昧な形状を入力することが許されていない。
- 形状の修正・変更が簡単に行えない。

われわれはこれらの問題に対する解答の方向として，形状特徴 (Form Feature) の概念に注目している。形状特徴とは，形状の部分的な特徴，例えば穴，溝，角丸め，切欠き，勾配などのことである。これらを形状表現の一部として明示的にもつとともに，形状入力や修正の対象として利用しようというのがその概念である。形状特徴は設計者が頭の中で形状を組み立てていくときの構成要素であり，かつ実際に物を作るときの加工の構成要素でもある。この点において，形状特徴の概念は現状の形状入力の概念に比べ，設計者により親しみやすいということがいえよう。さらにこの手法が有利な点は，形状の変更が容易である点である。例えば，穴の径や溝の位置を変えるとか，切欠きのもとの形状を変更したり，角丸めや勾配を変えるなどといった具合に，より柔軟に形状を修正していける可能性がある。さらにこの手法では，設計者の形状に対する意図がそのまま保持されていることから，CAM やシミュレーションなどの応用段階でのデータの利用においてもその有効性がいわれている。

また，3 次元空間内に図形を作成するための補助機能として，コンストラクション・プレーン (Construction Plane) 機能は有効であろう。これは，3 次元空間内の任意の位置にコンストラクション・プレーンを定義することで，このプレーン上に 2 次元形状を作成したり，それを深さ方向に操作したりできるという機能である。ディスプレイにはコンストラクション・プレーンを垂直に見るローカルなビューと，立体形状をダイナミックに見回せるアイソメトリックなビューとをオーバーラップさせ，複数のコンストラクション・プレーンを切り替えながら作業を進めていくのである。

おわりに

一般的な技術解説とわれわれが提供しているシステムの紹介めいた記述が混在し，読者に混乱を与えたのではないかと反省している。

われわれは本稿で，システム開発の中で知り得た情報をもとに，設計作業の中でコンピュータ・システムが果たすべき理想的な役割と現状の技術レベルとがどの程度なのかを少しでも読者に伝えたかったのである。

今後もハードウエアとソフトウエアの発展は続き，数値・文字データのみならず音声・画像データをも同列に扱うマルチメディア・データベース，数十 MIPS の処理性能をもつ EWS，4M ピクセルの高解像度ディスプレイと，とどまるところを知らない。

われわれはいつの間にかハードウエアを追いかけている状況に追い込まれているようにもみえるが，反対に，その気になればなんでもできる環境に今ようやくなりつつあるともいえる。われわれが夢想していたシステムをこれから現実のものとして提供できるようになると信じ，それに向かって努力している。

参考文献

1) 沖野教郎：「設計用コンピュータシステム構成論」，機械学会，Vol. 91, No. 833, 1988
2) 木村文彦：「形状モデリングと CAD/CAM」，精密工学会，Vol. 53, No. 3, 1987
3) 沖野教郎：「TIPS/GS で開く CAD の新世界グラフィックコミュニケーション」，日経メカニカル，1986. 1. 13
4) 「コンピュータ&ネットワーク LAN」，オーム社

● 本稿中特に注記はないが，AT&T 社，IBM 社，APOLLO 社，サン・マイクロシステムズ社などの各社の登録商標，および製品商標に対し，十分尊重していることをお断りしておく。

3次元機械設計用 CAD/CAM/CAE ユーザー調査

PIXEL編集部では，ユーザーが3次元機械設計用CAD/CAM/CAEをどのように認識し，利用しているか(利用しようとしているか)についてのアンケート調査を1988年11月から同年の12月にかけて実施した。以下にその結果を報告したい。3次元CAD/CAMの概要と最近の動向については，この特集の別稿を参照していただきたい。

この調査を通していくつかの興味深い結果が得られた。

- 使い方としてはデザイン・シミュレーション，ソリッド・モデリングによるCAE解析，ロボティックスなどのCIM指向が多い。
- CAD/CAM/CAEにおけるデータ互換では，異なるCAD/CAM/CAEシステム間の水平的なデータ互換とともに，CADからCAEやCAM，CIMやFAなどへの垂直的なデータ交換が不可欠になってきている。
- 3次元は使いにくいという認識が強いためか，使いやすいソフトウエアを望む声が高い。
- ソフトウエアとハードウエアもできることなら既成品で間に合わせたいという意識が強い。
- その場合のソフトウエアにはシステムの拡張性と自由度の高さ，カスタマイズの強化を望む声が高い。
- CADとCAMの連動は多くの場合これからであるが，データ交換ではなくデータを共有化したい，同一のシステムで処理したいという希望が大きい。
- 3次元システムの製図機能が低いのが問題という意見もあった。
- CAD/CAMのためのコンピュータについては，同じCAD/CAMソフトウエアを大型からワークステーションまで，いろいろな大きさのコンピュータで一様に使いたいという声が高い。
- ソリッド・モデリングについては，現在および将来は有効であるという意見が87%という高率の結果を得た。使いやすさやコンピュータ・パワーの問題などが解決されれば普及が見通せるであろう。
- パーソナルCADについては，ワークステーション以上のCADシステムとのデータ互換がキー・テクノロジになるであろう。

3次元システムが本格的に使われていくためには，まだ多くの問題が残されている。しかしCAD/CAMは使えるところから使っていくということが原則であろう。

現在の3次元機械設計用CAD/CAM/CAEの適用業務と今後の適用予定業務は何か？

●ロボティックス，ソリッドによる車体デザイン，流れ解析

現在は，CADとしてデザイン，外板，内板，シャーシ，コンポーネントなどの車両設計，エンジン設計，Die，工場内レイアウト，CAMとしてはプレス型，鋳造型，クレイモデル作製。CAEとしては構造解析，空力，クラッシュなどのシミュレーション作業工程に利用している。今後はCADデータをもとにした検査システム，機構設計などのメカニカルCADをもとにしたロボティックス，3次元サーフィスCADもしくはソリッドCADを利用する車のデザイン・システムに範囲を広げる予定。特にCAEとしては，構造解析，空力，クラッシュ，音響解析などは現在でもかなり行っているが，次のテーマとしては鋳物関係の流れ解析，車の外板関係のスタンピング時の縮み解析などがある。（自動車製造）

●デザインを強化

現在は自動車各部の機械設計，製図，デザイン・モデル作成やデザイン・モデルNC加工，FEMプリ・プロセッサ，電装設計に使用している。今後はデザイン・モデル能力を高める予定である。（自動車製造）

● CAMとCAEによる解析

現在は自動車用部品の設計に使用。また，ヘッドランプやリアーコンビ・ランプなどの照明器，ドアミラーやルームミラーなどの後写鏡の設計に使用。今後は対象品は先にあげた部品の設計とCAMの量の拡大とCAE解析業務への展開を図る。（自動車部品製造）

● CAM展開による加工の強化

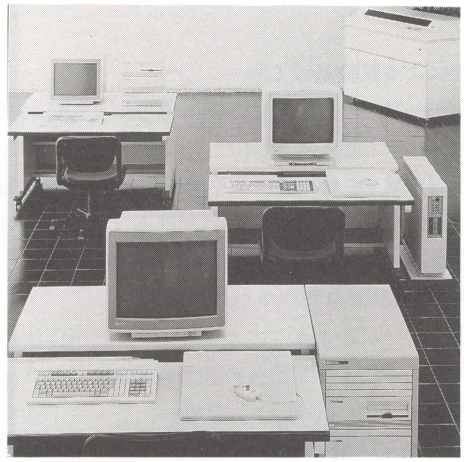
現在は自動車部品の干渉チェック，レイアウト検討，FEMモデル作成，ツール・パス検討業務に使用している。今後は設計検討業務，型加工や製品加工などのCAMへの展開を図る。（自動車部品製造）

●設計者1人に1台を配備，解析と意匠設計

現在は航空機関連の機械構造物の設計，干渉チェック，構造解析などに利用している。今後は各適用業務別にCAEを導入しパソコンCADによる製図を検討中。将来的には設計者1人1台のパソコンCADと解析専用，意匠設計専用などのCAEをファイル・サーバーを用いてネットワーク化する予定。（航空機部品製造）

●航空機のデジタル・モックアップ，設計・生産の一元化したデータベース構築

現在は航空機機体形状設計，NC機械加工部品の一部に使用。今後は配置検討図などのエリア・マネジメント，航空機機体のデジタル・モックアップ，機構解析，設計・生産用データベースの一元化のために機能拡張を予定。（造船・重機製造）



U-Graph（セイコー電子工業）

●ソリッドによる干渉チェック，意匠の検討

現在は産業機械設計・解析の際のFEMメッシュデータ作成，解析結果の表示に使用。今後は組立ての際の3次元ソリッド・モデルによる干渉チェック，3次元ソリッド・モデルにより重量慣性モーメントの求解，意匠デザインの検討への展開を予定。（産業機械製造）

● CAMへの展開

現在は缶形状の製品設計・解析，機械要素部品の強度解析に利用している。今後はCAMへの展開を検討中。（製缶）

●金型設計，NC加工，射出シミュレーション

現在は射出成形用プラスチック金型の製品部加工に3次元CAM，射出成形用プラスチック金型の図面作成に2次元CADを利用している。今後はプラスチック金型の設計，NC加工，射出成形シミュレーションなどのトータル・システムとして展開を図る。（金型製造）

●機構解析や動的な解析のシミュレーション

現在はプラント配管設計，石油掘削用ジャケットの設計，タービン設計，幕構造物（体育館）設計・プレゼンテーション，船舶設計，橋梁設計などに適用している。今後は構造解析用プリ/ポスト・プロセッサを充実させ，3次元モデルの有効利用を進めたい。また，機構解析などの動的解析を含めたシミュレーションにも取り組む予定。（プラント製造）

● 3次元プレゼンテーションを実用的に展開

現在はプレハブ建設物の設計用・施工図の作図，プレゼンテーション用の3次元グラフィックスに利用。今後は3次元のプレゼンテーションが実施レベルに載せられるようにシステム開発を行っていく予定である。（プレハブ建設）

●自由曲面による試作品づくり

現在は電子部品の2.5次元での設計・製図，CAMに利用。今後は試作部品の削り出しを早期にかつ正確に実現するために，2次元設計用CAD/CAMデータを活用した試作NC加工システム，またその試作を目的とした自由曲面などをもつ3次元形状部品の設計・製図へと展開したい。(電子部品製造)

● CAE機能の強化とロボットへの適用

現在はFEMなどのメカニカルCAEのモデル作成に利用している。今後はFEMなどのメカニカルCAEのモデル生成の機能拡張を図る。MOLDFLOWを用いたモールド金型製作用のCAEやCAD/CAMのフライスなどに利用したい。また，構造物のロボット・アームなどの干渉チェックや相関確認にも適用範囲を広げていく。(電気計測器製造)

●デザイン，NC加工へ展開

現在は家電製品・電気部品の意匠設計，NC加工データ作成に使用。今後は製品のデザイン，金型用CAD，NC加工データの作成の機能拡張を図る。(家電・電気部品製造)

現在使用中の3次元機械設計用CAD/CAM/CAEソフトウエアの構築方法は？

ソフトウエア・パッケージを購入	35%
ハードウエアと一体になったソフトウエアを購入	27%
自社開発	26%
社外に依頼	8%
ハードウエア・メーカーおよびソフトウエア・パッケージ・メーカーなどとの共同開発	4%

現在使用中の3次元機械設計用CAD/CAM/CAEシステムを今後拡張する場合，どのような方法を取るか？

ソフトウエア・パッケージを購入	35%
ハードウエアと一体になったソフトウエアを購入	27%
自社開発	20%
ハードウエア・メーカーおよびソフトウエア・パッケージ・メーカーなどとの共同開発	15%
社外に依頼	3%

現在と今後とを比べてみると，ソフトウエア・パッケージの購入とハードウエア／ソフトウエア一体となったいわゆるターンキー・システム購入が多いことは変わらず，また自社開発がやや減少し，ハードウエアおよびソフトウエア・メーカーなどとの共同開発が増える，オーダーメードのソフトウエア開発委託は減少という結果になっている。

現在使用中の機械設計用CAD/CAM/CAEシステム構築のためにかかった費用は？

●ハードウエア，ソフトウエア込みで50億円以上(自動車製造)

●ハードウエア，ソフトウエア各10億円，合わせて20億円(自動車製造)

●ハードウエア，ソフトウエア込みで約10億円（造船・重機製造)

●ハードウエア，ソフトウエア込みで8億4,600万円（自動車部品製造)

●ハードウエア，ソフトウエア各3億円の計6億円（家電・電気部品製造)

●ハードウエアが約3億円，ソフトウエアは約5,000万円の計約3億5,000万円（プレハブ建設)

●ハードウエアが2億2,000万円，ソフトウエアが5,000万円の計2億7,000万円（製缶)

●ハードウエア1億円以上，ソフトウエアが5,000万円以上(自動車部品製造)

●ハードウエア，ソフトウエア込みで1億5,000万円（航空関連製品)

●ハードウエアが9,000万円以上，ソフトウエアが4,000万円以上（電気計測器製造)

●ハードウエア（EWS）とCAEソフトウエアで1,000万円。ホスト・コンピュータと2.5次元CADソフトウエアを加えて1億2,000万円（印刷機械製造)

●ハードウエア，ソフトウエア込みで1,500万～2,000万円(金型製造)

金額については各社で要求仕様や含まれているもの（特に人件費）などが異なり，時期や期間も違うため，目安程度と考えていただきたい。

3次元機械設計用CAD/CAM/CAEシステムを構築・選択する際の注意点は？

●他のシステムとの互換性，システムがその分野においてメジャーなものか

①ソフトウエア・パッケージを購入する場合，現在所有の他システムとの互換性，サポートの良し悪しなどのベンダーの信頼性，ソフトウエアが要求する機能を備えているかどうか，使い勝手の良さ，拡張性に注意した，②要求する機能をもつソフトウエア・パッケージが市販されていない場合，社内開発を行った，③採用予定システムがその分野においてメジャーなものかどうか，そのシステムの位置付け，システムを使用する予定

の分野における位置付けなど，例えば鋳鍛造業界ではそのシステムが広く使われているかどうか，またカー・デザイン分野ではそのシステムがメジャーなものなのかどうかという点。(自動車製造)

●使いやすさ，CADデータの互換性

①初心者に親しみやすく，使いやすいシステムということ，②3次元設計・検討はCADの方が圧倒的に速いが，手作業と同等以上の2次元製図能力をもつということ，③CADデータが社内のNCシステム，FEMシステムで使用できるものとした。(自動車製造)

●自由曲面，ネットワーク化，バージョン・アップ

①機能上は，自由曲面のハンドリングが容易なこととNC機能が豊富で，システムの拡張性，ネットワーク化が容易なこと，②バージョンアップなどのシステム開発力が高く，サポート体制がしっかりしていること。(自動車部品製造)

● 2次元CADとのデータ互換，ソリッドへの拡張

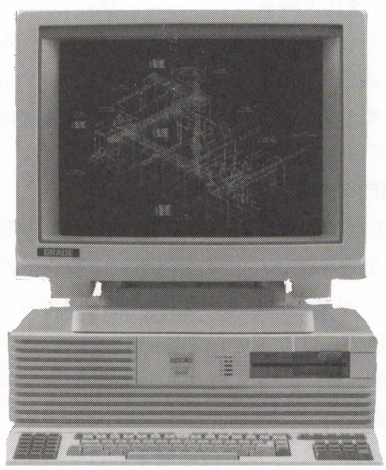
2次元CADとのデータ互換性，ソリッド化など将来への拡張性，システムの実績，コストの問題など。(自動車部品製造)

●将来の拡張と処理の高速性

多機能で将来，拡張性が十分にとれるもの，高速処理が可能なことに注意してシステム構築した。(航空関連製品製造)

●業務提携先とのインタフェース

海外，および国内の同業他社とのインタフェースを考慮した。



GRADE/P(日立造船情報システム)

(造船・重機製造)

●ハードウエアは標準品，ソフトウエアはコスト・パフォーマンス

EWS上のCAEシステムの場合，ハードウエアは標準的なもの，ソフトウエアはコスト・パフォーマンスの点に注意して構築。ホスト・コンピュータ上の2.5次元CADシステムの場合，標準的なシステムとして一般的に広く使用されているもの，レスポンスの良さとシステムの拡張性など。(印刷機械製造)

●データベースの一元化，カスタマイズ，自由曲面

データベースの一元化ができること，早期に基本的なカスタマイズと標準部品の登録処理が実行できること，SEの養成，ソフトウエアの一元化，操作性が良いこと，ハードウエアが3次元に対応できること，コスト・パフォーマンスが高いこと，自由曲面が完全にサポートされていることなど。(金型製造)

● CAM，他社での使用実績

CADだけではなくCAMへの展開ができること，同業他社での使用実績，作図作業のしやすさ，応答性の良さなど。(金型製造)

●ユーザー・インタフェース，システムの拡張性，トラブル時のサポート

システムの機能と操作性，ユーザー・インタフェースの容易性，システムの拡張性と開発方針，価格，トラブル時のサポート体制などの点。(プラント製造)

●パッケージの場合，拡張性と自由度

パッケージ・ソフトウエアを購入してシステムの構築を行ったが，パッケージ・ソフトウエアの性能を十分生かしながら当社独自の要求に合ったシステムを作り上げることを目指して開発を行った。パッケージ・ソフトウエアの選択時には第一に拡張性の高いもの，絵を描くうえでの自由度の大きなものを選択した。(プレハブ建設)

●他のCADデータとの互換性，システムの拡張性と使いやすさ

CAE，他のCADとの互換性などのシステムの拡張性と使いやすさという点。(家電・電気部品製造)

● 3次元のモデル生成機能

まず要求仕様にかなう機能をもつ3次元モデル生成用ソフトウエアの選定を行い，そしてそのソフトウエアの機能を十分に生かすことのできるハードウエアの選定をしてシステムを構築。(電気計測器製造)

●カスタマイズ，処理速度

業務に適合させるカスタマイズ機能が充実しているか，機能開発が容易に行えるか，システムの基本機能が充実しているか，システムのレスポンスは実用的な速度をもっているかの点に注意してシステムの構築を行った。(電子部品製造)

3次元機械設計用CADとCAMの連動は？

現在連動している	50%
システムとして結合しているがまだCAMは動いていない	19%
CADの機能しかない	13%
CAMの機能が弱く実務では使えそうにない	6%
その他	12%

CADとCAMの連動については？

● 3次元サーフィスではCAMと連動，加工のことを設計者がよく知らない?

3次元CADの分野をいくつかに分けて考えている。まず，3次元ワイヤー・フレーム・モデルCADではCAMとの連動は考えていない。しかし，キネマティックスを使用する場合はロボティックスCAMとの連動がある。次に，3次元サーフィス・モデルの場合は，CAMと連動することにより作業の日程短縮，コストセービングに効果があり，収入を得られるシステムとなる。しかし，現在CADを使っているエンジニアが加工のことまで考えてモデリングしてくれればよいのだが，あまりそのようにはしてくれない。現場での加工方法をエンジニアが知らないということもあるようだ。(自動車製造)

●精度の高い共通フォーマットが必要

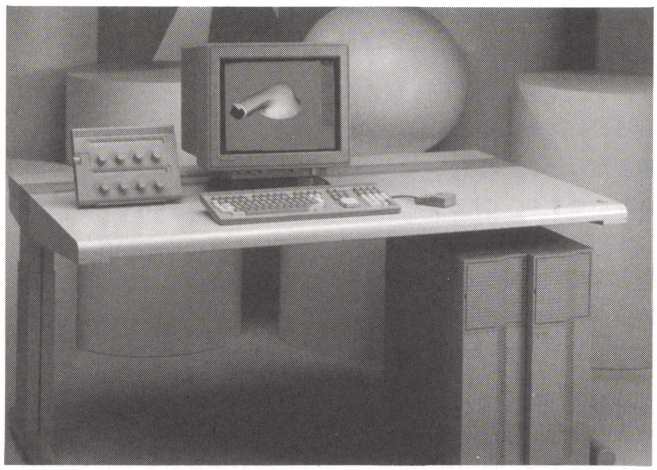
CADとCAMの目的は本来異なったデータ群を取り扱うことである。CADは形状データからなり，CAMは加工方法のノウハウ体系からなる。しかし，CAD上の基本形状を利用すればCAM化のミス防止と作業工程の効率化が図れそうであるので，ソフトウエア上の工夫があれば可能性は非常に大きいものがある。ユーザーとしてソフトウエア作成者に望みたいのは，2次元のIGESでさえあまり互換性がないが，3次元を取り扱うにはデータが命となる。互換精度の高い共通フォーマットを決定して市販を行ってほしい。(自動車部品製造)

● CADとCAMのバランスのとれたシステムが少ない

CAD/CAMの双方のバランスのとれたシステムが少なく，そのどちらかが優れているシステムの場合が多くて困る。また，今後は生産情報なども関連をもてるようなシステムができることを望む。(家電・電気部品製造)

● 2次元製図，3次元サーフィス，ソリッドを1つのシステムで

2次元で製図，3次元サーフィスで形状とカッター・パス，3次元ソリッドで解析を行うので，リンクなどというよりも1つのパッケージの中ですべての作業のできるシステム，またはすべてのワークステーションで同時にすべて同じ作業をする必要はないので，なんらかの作業が効果的にできるシステムがよい。(金型製造)



自由形状設計加工システム/フレスダムFRESDAM(ソニー・テクトロニクス)

● CADとCAMが同一システムで

同一システム内でCADとCAMが稼働しなくては，モデル製作，型設計，NCデータ作成の全部門で連携をとりながら業務効率の向上を図ることは難しい。現在はこのような条件を満たしたシステムを作成し，使用している。(自動車製造)

● CADとCAMの連動は不可欠

CAD/CAMが連動していないと効果がでないし，システムを導入しても作業工程などの仕組みも変わらない。(自動車部品製造)

● CAD/CAM一体化で顧客のニーズに迅速に対応

CADの作図データはCAMにおけるデータベースだと考えているので，多品種・少量生産に対応していくためにはCAD/CAMを一体で考えてなければ顧客のニーズに迅速に対応できないと思う。(プレハブ建設)

●設計と製作部門の連動が必要

CADとCAMは連動した方が望ましいが，その連動のためには以下のような問題点もある。①設計部門と製作部門の組織を含めた体制づくりが必要である，②同一部門であるか，そうでないのかでCAD/CAMの連動がスムーズにいかない場合がある，③設計，技術，製作の各部門の連動が組織的に必要である。(造船・重機製造)

●ソリッド・モデル・レベルでの連動はこれから

CAMの中で鋳造の型については実用化の段階にあると考える。しかし，3面図から加工情報を取り出し，ソリッド・モデルに変換してから加工情報を付加するほど，3次元機械設計用のCADとCAMの連動が進化しているとは思わない。(産業機械

製造)

●加工時では3次元データを使用していない

いまだ実用化していない状況での感想としては，第1に当社製品がモールド型などの3次元加工の必要性が低い。第2にモールド型も多用しているが，ほとんどが設計・製造とも行いやすいように直線，円，斜線の合成による単純形状にしているため，加工時には2次元データにバラして使用している。(電気計測器製造)

● 2.5次元データで現状は十分

3次元的な形状はほとんどないので，2次元CADにアプリケーションでZ軸を追加して2.5次元のデータとして利用している。今のところはこれで十分である。(印刷機械製造)

● CAMと連動しなければ投資効果がでない

CADだけでは投資効果が十分ではないのではないか。CAD/CAMでトータルとして効果がでるのでは?(製缶)

● CADシステムにはCAMが，CAMシステムにはCADがそれぞれ中途半端な機能しかない

CADからみたCAM，CAMからみたCADが付属的存在であり，利用者にしてみれば中途半端な機能しかないために，結局CAD→CAMというシステムを構築せざるを得ないのが現状である。今後は，CADはCADの機能を，CAMはCAMの機能をというようにそれぞれ独立に熟成させ，各システムで相互の切り口を用意したものが必要となってくる。すなわち，分野別専用化システムと授受情報の明確化が要求される部分である。(航空関連製品製造)

● CAM技術者のノウハウをデータやカスタマイズ機能として盛り込めるシステムが必要

CAMのノウハウはCAD利用者のもつ技術と異質の技術であり，CAD利用者がCAM技術を駆使することは容易ではない。したがって，CAM技術者のノウハウをデータやカスタマイズ機能として盛り込めるCAD/CAM機能が重要となり，CAD技術者とCAM技術者の接点でインタフェースの役割を果たすことがCADとCAMの連動の最大のポイントとなる。(電子部品製造)

3次元機械設計用CAD/CAM/CAEメーカーのユーザーに対するサポート体制は？

あまり良くない	53%
良い	13%
普通	13%
悪い	7%
その他	14%

サポート体制がよくないとされる理由は？

- 実際の機械図面は3面図どころか，2面図＋断面形状である。これを情報源にNCテープが作成できるところまで一元化したシステムをもっているベンダーがいない
- メーカーは機能アップよりも新しいハードウエアに移植することに力を注いでいる
- 個別質問に対しては回答が遅い。長期的にみれば解決されるが，現在のトラブルの解決がなされない
- バージョンアップ後の他システムとの互換性が悪い
- CAD/CAMベンダーがユーザーの仕事をよく知らないということと，ユーザーも企業秘密のためCAD/CAMベンダーに話せない内容がある
- メーカーやベンダーに技術の高い技術者が少ない
- 外国製ソフトウエアのためか，バージョンアップやソフトウエアのバグに対する対応が遅いし明確ではない
- ソフトウエア開発部署と機能の使用部署が異なるため，実用化へのステップアップが必要で，すぐに使えない状況になっている

現在使用中の3次元機械設計用CAD/CAM/CAEシステムの機能についての不満や改善点は？

●得意な分野をもっていること，EWSだけでなく大型コンピュータでも実行できること

サーフィス機能追加などのユーザー・デファイナブルな機能を強化してほしい。万遍なく機能強化を図るのも一つの方法だと思うが，得意とする分野は少なくともきちんとした機能を提供してほしい。例えば，CAMにおけるポスト・プロセッサでサポートするNCフォーマットの種類など，また2次元から3次元，バージョンアップ時のアップグレード・コンパチビリティなど，同一ベンダーもしくは同一CAD/CAMシステム内でのデータ交換はきちんと行ってほしい。それから，すべての処理をEWS上で実行するのはよいが，やはり処理に馬力を必要とする計算は，それなりの大型計算機で実行できるようなかたちにシステムを設計してほしい。(自動車製造)

● CAEはEWS，CADはメインフレームとメーカーが異なるためデータ交換が面倒

CAEはEWS，CADはメインフレームというようにメーカーが異なるため，データ交換が面倒である。ソフトウエアが別々のメーカーのため，結合するためのソフトウエアは自社開発しなくてはならない。ソリッド・モデラーの操作性が良くなく，FEMモデルを作成するために導入したのだが，通常のプリ・プ

ロセッサで作成する方が楽に思える。ソフトウエアの価格とサポート料が高いのにバグが多い。(印刷機械製造)

●データベースが重い，日本語化がまだ

管面上でのメニューチョイス方式を採用していることと，データベースが重いためオペレーション・スピードが遅い。日本語化されていない。(自動車部品製造)

●他のシステムとのデータ互換ができない

他システムとデータの互換性がほとんどない。例えば，IGESなどのフォーマットへのインタフェースを標準としてほしいが，IGESだけでは用をなさない。(電子部品製造)

●ホスト・コンピュータ使用のため端末数の増加に伴い待ち時間が長くなる

ホスト・コンピュータでデータベースI/O，図形処理集中演算をやらせているが，端末数の増加につれて待ち時間が増加している。このため来期より高速のGWSタイプのシステムに置き換えていきたい。(自動車製造)

●扱えるモデルサイズが小さい

システムのレスポンスが悪く，扱えるモデルサイズが小さい。コストが高く，オペレータの教育に時間がかかる。(造船・重機製造)

● CAMが自動プロのため効率が悪い

3次元CAMは自動プロなので，オペレータの個人差が大きく現れる。また，部品図を製図してそれからNCデータを作るので効率が悪い。紙テープによる運用なので手間がかかるし，CPUが16ビットなので処理能力が悪い。(金型製造)

汎用設計援助システムUNICAD(日本ユニシス)

● CAM加工での完全な形状定義が無理

CAMへリンクした場合，データ不足が目立ち人力による補充が必要である。3次元といえどもソリッドではないものが多く，CAM加工における完全なる形状定義が無理。(自動車部品製造)

●製図機能が弱い

3次元CADソフトウエアであり，製図機能，特に図面として出力する機能が弱いので，ソフトウエア付属の簡易言語によるプログラムでサポートしたい。(製缶)

●サーフィスのため干渉チェック，解析などが難しい

現在のシステムはサーフィスまでの機能で干渉チェック，解析などに対して作業量が多い。また，干渉チェックなどは画面上では判定しにくい。処理速度が遅い。(航空関連製品製造)

● CADデータからCAMデータを取り出すのに手間がかかる

当社で使用しているパッケージ・ソフトウエアは，絵を描くうえではいうことはないのだが，CADの絵からCAMのデータを取り出すのに非常に手間がかかる。(プレハブ建設)

●自由曲面の設計・加工が困難

基本的に2.5次元のシステムで，面の機能をもたない。したがって，自由曲面の設計・加工が単独では困難である。(電子部品製造)

現在運用中の3次元機械設計用CAD/CAM/CAEシステムは予定通りの成果を上げているか？

成果を上げている	50%
成果を上げていない	29%
その他	21%

成果を上げていない場合の理由は？

●一部の部門だけの適用では効果が上がらない

設計者が2次元的思考に慣れているため，オペレーションの習慣がない。特定部門だけの適用ではなく，全社的にトータルに取り組み，適用を図らないと3次元システムの効果は上がらないが，そのための体制づくりや現状システムを変えることに対するコンセンサスを得るのが非常に困難である。(プラント製造)

●解析などができる一方で導入効果が図りにくい

図面として出力する機能が弱い。解析ができるので質の良い設計ができると思われるが，設計時間の短縮という意味では効果を把握しにくい。(製缶)

投資の割に効果が把握できない。(産業機械製造)

●現在では要求仕様が満たされていない

要求仕様に対応するシステムが構築されていない。3次元設計手法，統合データベース，高速通信網などの整備が必要である。(造船・重機製造)

● CAM導入の効果はすぐにでない

CAMは相当の訓練を要し，すぐには効果がでない。(自動車部品製造)

3次元機械設計用CAD/CAM/CAEのソフトウエア・パッケージに対するユーザーの期待は？

●操作が容易で，いろいろなプロセスが高速に取り扱える

まずオペレーションが容易でモデリングから解析までのプロセスが速いこと。断面形状・要素特性などの属性情報を容易に取り扱えること。他機種とのネットワークが組めることなど。(航空関連製品製造)

●同一のソフトウエアがいろいろな階層のコンピュータで使える

ユーザー部門でももう少し手軽に使えるように操作性が良く，メインフレーム—EWS—パーソナルコンピュータなどのように階層的に使えるシステム構成がとられ，処理の分散／データベースの集中・一元化が可能なもので，CAD/CAM/CAEのみでなくCATの機能も有するもの。(造船・重機製造)

●通常は2次元システムとして使え，必要なときに3次元機能に拡張できる

通常は操作性の良い2次元CADとして利用でき，必要なときに2次元からの拡張として3次元モデルを作成できて，なおかつソリッド・モデラーとしてFEM/CAMへの展開ができるもの。慣性モーメント，重心位置，接続の判定などの解析機能のあるもの。最終的にはプロダクト・モデルを扱うための十分な機能をもつソフトウエアを望む。(印刷機械製造)

●他のCAD/CAMシステムとのデータ互換性と単一のアーキテクチャ

他のCAD/CAMシステムとのデータの互換性をもち，操作もデータも単一アーキテクチャをもつソフトウエアで，生産情報などのDP部門とのデータ互換が図れ，なおかつバージョンアップ時のカスタマイズの簡単なものを希望する。(電子部品製造)

●ユーザーがカスタマイズを簡単にできる

マルチベンダー対応で，ユーザー側のカスタマイズが容易で，システムの発展性／将来性が有望なソフトウエアを希望する。(電気計測器製造)

● 2次元製図もソリッド・モデルの作成もできる

操作性の良いソフトウエアで設計から加工まで行え，強力な2次元作図機能とサーフィス機能をもち，ソリッド・モデルに対応しているソフトウエアを望む。(金型製造)

●コンピュータ・グラフィックスとの連動

ソリッド・モデルが容易に扱え，CGとCAD/CAMとの連動が容易になるデータベースをもつソフトウエアを期待する。(自動車部品製造)

●インテリジェント機能でユーザーの負担軽減

人工知能型CAD/CAM/CAEのような利用者の負担を減らすものがよい。(造船・重機製造)

●日本語と3次元データの規格化

日本語規格と3次元データの規格の統一がきちんと図られたソフトウエアを希望する。(自動車部品製造)

● 3次元ソリッド・モデルの作成

2面図＋断面情報から3次元ソリッド・モデルを作成できるソフトウエアを望む。(産業機械製造)

● 2次元情報から簡単に3次元モデルができる

2次元で描かれた図面を3次元で手軽に描くことができ，またCAMデータに手軽に落とすことができるソフトウエアが欲しい。(プレハブ建設業)

ユーザーが期待する3次元機械設計用CAD/CAM/CAEシステムの形態は？

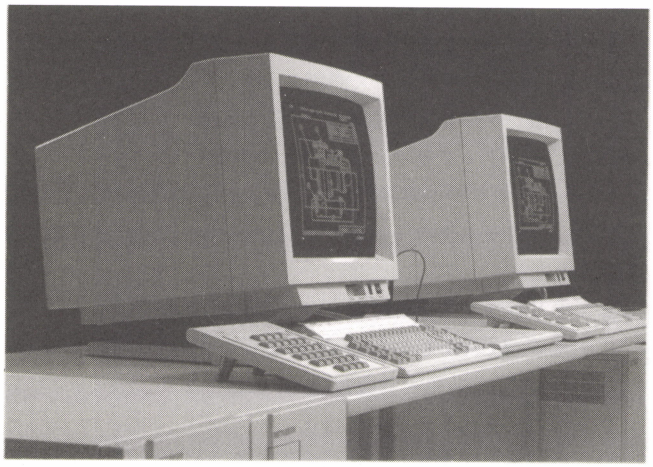
ワークステーションによるパーソナルなシステム	48%
スーパー・ミニコンによる小型システム	19%
パーソナルコンピュータによる低価格なシステム	14%
大型コンピュータによる大規模システム	9%
その他	10%

3次元機械設計用CAD/CAM/CAEシステムにユーザーが今後求めるハードウエア

●低価格で高速の3次元グラフィック・ディスプレイ。低価格でLINPACK値2MFLOPS以上の倍精度浮動小数点演算能力をもつワークステーション。低価格で大型コンピュータと同等の速度のデータベースI/O能力をもつワークステーション（自動車製造）

●コンピュータ・パワーが良くて信頼性が高くシステム構成が柔軟であり，高解像度のCRTで3次元対応可能なもの。例えば，YHP社のHP 370 SRX（金型製造）

●解析&CGでレスポンスの速いもの，CRAY並みの処理速度をもつワークステーション（自動車部品製造）



MELCAD-MD システム(三菱電機)

- 2,000 万円程度でシミュレーションがリアルタイムで可能なミニ・スーパーコンピュータと，複数のユーザーが同時に操作できる環境を小型システムで可能とするハードウエア（産業機械製造）
- UNIX も MS-DOS も走る EWS で低価格なもの（印刷機械製造）
- 異機種のハードウエアを簡単につなぐことができ，管理が非常に簡単でしかも計算速度の速いハードウエア（プレハブ建設）
- ホスト・コンピュータ並みかそれ以上の能力とカスタマイズ機能をもち，Ethernet などによってネットワーク化できる EWS（電子部品製造）
- 低価格で高速処理の可能な高解像度ディスプレイ（家電・電気部品製造）
- ワークステーションのレスポンス・タイムの速いもの。高速で安価な通信が可能なシステム（造船・重機製造）
- マルチタスクは当たり前。低コストで通信機能のスピードアップを図ったハードウエア（自動車部品製造）

ソリッド・モデルについては？

現在の業務には有効ではないが，将来的に有効	47%
現在の業務内容に対して有効	40%
現在，将来とも効果的ではない	0%
その他	13%

3 次元機械設計用 CAE については？

現在の業務内容に対して有効	67%
現在の業務には有効ではないが，将来的に有効	26%
現在，将来とも効果的ではない	0%
その他	7%

3 次元機械設計用 CAE の導入については？

すでに導入済み	53%
導入を具体的に進めている	27%
導入を前提に調査・勉強中である	13%
導入の予定はない	7%

3 次元機械設計用 CAE の適用事例としては？

- 部品干渉チェック，たわみ／歪み・変形シミュレーション，共振・固有値の振動解析，形状を小さく抑えるための重量の削減と材料費削減のための強度シミュレーション（家電・電気部品製造）
- 主に FEM を使用した解析が有効で，工数のかかるモデル作成を CAD で行っている。CAE の適用部品としてはピストン，コンロッド，クランクシャフトや二輪車体の大物部品がある（自動車製造）
- 光学設計シミュレーション，オンマウントの振動解析，樹脂流動解析（自動車部品製造）
- 航空機の脚および扉の機構解析，航空機諸元検討プログラム，数値風洞・構造解析に使用（造船・重機製造）
- 射出成形のシミュレーション，金型内の冷却回路の検討，射出圧による金型の変形シミュレーション（金型製造）
- レイアウト検討，組立性・加工性検討，実モデルによる設計評価，FEM モデル作成（自動車部品製造）
- FEM のプリ/ポスト処理（産業機械製造）
- 構造解析，空力シミュレーション，クラッシュ時のシミュレーション，音響解析，メカニカル・アナリシス（自動車製造）
- 計測機センサー類や制御機器／筐体類を対象とした構造解析，振動解析，電磁場解析，流体解析（電気計測器製造）
- ジャケットの 3 次元モデルから自社開発プリ・プロセッサを介して構造解析ソルバーとリンクしている（プラント製造）
- FEM，重心，慣性モーメント（GD^2）の解析。製図は 2.5 次元の別 CAD システムで行っている（印刷機械製造）
- ユーザー言語による半自動設計システムの構築（パラメータ設計）

●イメージデータ→フォトデータ・システム構築。2.5次元NCシステム開発（フライス）
●ユーザーCADとの図面交換。簡易デジタル・パターン設計機能開発（電子部品製造）

パソコンを中心とした3次元機械設計用パーソナルCAD/CAMシステムについては？

ハード，ソフトともに現在の業務では使えない	26%
価格の割に良くできているが業務では使えない	26%
ソフトは良いがハードが貧弱	16%
ハードは良いがソフトが非充実	11%
ハード，ソフトともに十分に現在の業務で使える	6%
ハード，ソフトの機能は良いがサポートの問題が心配	5%
よく知らない	5%
その他	5%

パソコンを中心とした3次元機械設計用CAD/CAMシステムのもつ問題点

●上位システムとの互換性

今後の増設に対して安価なパソコン・ベースのソフトウエア（製図機能のみ）を検討したいが，ソフトウエアが異なるとデータベースの変換がうまくいかないようである。IGESについても図面形状のみの互換レベルだとの話もある。上位システムと互換性のあるパソコン・レベルでの安価なCADソフトウエアが望まれる。（製缶）

●処理速度が遅い

現在のパソコン・ベースのCADはかなりレベルが上がってきたと思うが，EWSなどと比較すると処理速度という点で問題がある。3次元設計を行う場合，機能，オペレーションの面でまだまだ制約が多い。ただし，価格面まで考慮するとこの点においては勝っていると思う。（航空関連製品製造）

●ソフトウエアとデータのメインテナンス，大型CADとのデータ互換，パーソナルコンピュータの信頼性

ソフトウエアやデータが分散するためのメインテナンス・管理上の問題と，小メーカーが多いため特に大型CADなどの他システムとのデータの互換がないこと，そしてパーソナルコンピュータやハードウエアの信頼性が低いことがあげられる。（電気計測器製造）

●扱えるデータ量が少ない

データ量の少ない小物部品の設計には有効だが，データ量の多い図面（A2サイズ以上）では処理能力不足のため，自動車製造の分野では手作業以上の能率を上げることは難しいと思われる。（自動車製造）

●データの精度，ネットワーク

基本的にはよくできているが，突っ込んで使うとデータの精度，図形および寸法線の編集，パラメトリック図形自動トリミング，ハッチング，入力方法の自由度，カスタマイゼーション，ネットワークなどの制約がある。（金型製造）

●NC機能，CAD→CAM→CAEへの展開

NC処理時のコンピュータ・パワー，OSの違いによるデータのポータビリティ，データの集中管理，例えばCAD→CAM→CAEのようなデータの応用，以上の4点の問題がある。（自動車製造）

●モデリング機能，コンピュータ処理能力

モデラーの機能不足，メインフレームCAD/CAMとのデータベースの一元化の問題，CPU処理能力の不足，操作性の問題がある。（プラント製造）

●ソフトウエアの拡張性

一般的なCADシステムとのデータの互換性，ハードウエアの性能，ソフトウエアの拡張性などの問題がある。（印刷機械製造）

●周辺機器

CPUパワーは良いレベルに達しているが，周辺機器がまだまだ弱い。（プレハブ建設）

●複雑な形状のモデリング

単純な機械部品を扱うのが限界で，複雑な構成品が表現できない。（産業機械製造）

現状の3次元機械設計用CAD/CAM/CAEシステムの問題点は？

●CAD→CAM，CAD→CAEの流れにおけるデータの授受

CAD→CAM，CAD→CAEの流れを考えた場合，データの授受に問題がある。例えば，CAEにおけるプリおよびポスト・プロセッサにデータを渡そうとした場合，中間フォーマットがそれぞれ異なっている。CAEを考えた場合，1つのプリ/ポスト・プロセッサでNASTRANやMARK，ABACUSやSAPなどのすべては取扱いができず，それぞれのソルバーごとに適したプリ/ポスト・プロセッサが必要である。CAMを考えた場合は，実際のわれわれの仕事の手順に合うように設計されていない。3次元CADはまだまだ2次元CADと比較すると安定せずにエラーが多い。メーカーは営業の人員を強化するのと同様にSEも強化してほしい。中には図面の読めないSEもいる。ユーザーが独自の機能追加を行う場合があるが，この場合FOR-

TRAN でコーディングする CAD/CAM システムがまだある。全部の FORTRAN をなくせとはいわないが，少なくとも CAD/CAM のコマンドをそのまま使えるような機能にしてほしい。(自動車製造)

●ソリッドとサーフィス間のデータの互換性，モデラーとソルバーの間のインタフェースの標準化

ソリッドとサーフィスの考え方が異なりデータの互換性がなく，ほとんどのシステムが独自のデータ構造をもつために，異機種間でデータのやり取りをするとモデルの精度が落ちる。また，用語も統一されていないためシステムが異なると操作しづらい。B-Spline，ベジエ，クーンズなどの面の考え方を統一または整理し，ソリッドとサーフィスのデータの共有化を図り，モデラーとソルバーの間の標準インタフェースの整備が期待される。CAT やロボティックス，キネマティックスなどへのシミュレーション分野の拡張が必要。また，3 次元測定機器からのデータをモデリングに利用できるような入力システムの開発が必要である。(家電・電気部品製造)

● 3 次元でなく 2.5 次元 CAD でもかなりのことができる

2 次元 CAD に比べて 3 次元 CAD は操作性が良くなく，スピードが遅い。3 次元で設計したとしても加工側がそのデータを利用できなければ何の意味もない，結局 2 次元図面にしなくてはならないので一貫システムにしなければメリットはでない。当社でも CAE で FEM モデル用ソリッド・モデラーを導入したが，操作性があまり良くない。もっとやさしく取り扱えるようにならなければ 2 次元の CAD のように使えない。2.5 次元を 3 次元 CAD とよぶのであればその CAD は当社でも使用しているが，2 次元 CAD としてしか利用していない。しかし，十分なアプリケーションがあれば 2.5 次元 CAD でもかなりのことができると思う。CAM などでは 2.5 次元を 3 次元に立ち上げて自由曲面などが生成できるものもあるようなので，CAM としてはこちらの方がソリッド・モデラーを使用するよりよいように思われる。(印刷機械製造)

● 3 次元 CAD の 2 次元作図機能，2 次元と 3 次元でメーカーやハードウエアが別々

CAD/CAM で DNC 加工するにしても組立て，修正，改造，打合せはすべて人手によるものなので，質の高い 2 次元の図面への要求は高い。現状の CAD は 2 次元用と 3 次元用とで開発が別々の場合が多く，それぞれに一番良いものを選ぶとソフトウエアおよびメーカーが別々で，そのうえハードウエアまで別々になる場合が多い。同じ人間が作業内容によってこれらを使い分けるのでは効率が良いとはいえない。多くの 3 次元 CAD の 2 次元作図機能はあまり進歩していないように思う。また，操作性が悪いものも多い。すべての CAD にいえることは，表示画面が小さく，大きな図面を原寸で全体が見られない。37 インチの CRT を使うわけにもいかないので，CRT を 2 台サポートするかスクロール機能をつけてほしい。(金型製造)

●操作の複雑さ，レスポンスの悪さ，データベースのボリューム

2 次元→ 3 次元時のオペレーションの難解さと作業負荷の増大，2 次元に比べたときのレスポンスの劣化，データベースのボリューム，他システムとのデータ交換の問題を改善する必要がある。(電子部品製造)

●入力の難しさ

3 次元の入力後の多用性と有効性は理解しているつもりだが，入力がいかにも難しそうにみえる。これまでの手書き製図技法が身についた設計者には現状の入力方法はいずれも違和感があり，簡単ではないという偏見が全般的にありそうだ。(電気計測器製造)

●マン・マシン・インタフェースの向上

当社では建築図面を描くため対話式でタブレット入力をしているが，やはりタブレットは鉛筆の代用品でしかなく，せっかく高い CAD を導入しているのだからもっと高機能の入力方法がないかと常に考えている。そのような意味から CPU パワーが大きくなるにつれてマン・マシン・インタフェースの向上が問題となってくると思われる。(プレハブ建設)

● CAM 関連ソフトウエアの不備

検図業務の際，相関が理解しにくい。本格ソフトウエアは高額過ぎる。CAM 関連ソフトウエアの不備が多く，80%以上は自社内でソフトウエア開発が必要。(自動車部品製造)

InterAct—32C シリーズと 300 シリーズに用意されたデュアル・スクリーン・ワークステーション(武藤工業)

BIBLIOGRAPHY

パソコン画像処理

編／テレビジョン学会
著／花木真一，岩下正雄，寺嶋廣克
発行所／昭晃堂
☎ 03(269)3449
初版発行日／昭和 63 年 9 月 20 日
定価／2,700 円
判型／A5 判
ページ数／212 ページ

ここ数年のハードウエアの発展に伴い，パーソナルコンピュータも 8 ビットから 16 ビット，最近では 32 ビットまで多くの機種が発表されている。パーソナルコンピュータが発表される以前は一部の専門家だけのものであったコンピュータも一般に広く利用されるようになり，従来は非常に大量のメモリを必要とした画像処理の分野においてもパソコン画像処理という一分野が確立されつつある。

本書は，テレビジョン学会誌に講座「パソコンによる画像処理」として連載したものをベースにまとめ直したものである。

本書のねらいとして，パーソナルコンピュータ程度で手軽に画像処理を行いたいという読者の手引きとなるような情報の提供，デジタル画像処理の手法を理解するための入門向けでしかも背景としてパソコン環境を強く意識するような情報の提供を中心に，パソコン画像処理の実現形態と特色，効果，現状，今後の発展に関心をもつ読者に対して情報の提供を考慮してまとめられている。

企業の研究所に属し，画像処理，パターン認識，画像処理プロセッサ，マン・マシン・インタフェースなどの最新の実務的な研究を重ねている著者らの豊富な経験が，実例として多数掲載されている。研究所での最新事例を中心に，大学や他の研究機関から学会で発表されたテーマまで幅広く取り上げている。

これから画像処理に取り組もうと考えている方にとって画像処理の入門書として最適である。

本書は 9 章からなり，各章は次の通り。◎緒論◎ディジタル画像処理◎パソコンの発展と画像処理◎画像処理システム◎画像処理の基本手法◎ソフトウェアとインタフェース◎コンピュータグラフィックス◎動画像とビデオ処理◎画像処理応用事例

CAE　新製品開発・設計支援　コンピュータ・ツール

編著／相澤龍彦，前川佳徳
発行所／共立出版
☎ 03(947)2511
初版発行日／昭和 63 年 6 月 10 日
定価／2,800 円
判型／A5 判
ページ数／226 ページ

CAD/CAM が産業界に普及してきた今日，試作品を数多く作らなくともコンピュータ上で設計物を構築し，シミュレーションなどのテストを行う CAE に注目が集まっている。試作品の製作工数を短縮できる CAE には，開発期間の短縮化および高精度化が要求される，現在の激烈な開発競争に打ち勝つための期待がかけられているのである。

同書の冒頭にもあるように，CAE はこのように実用問題に汎用的にこたえ得る技術として展開され，特に作今の高機能・高品質を要求される新製品開発および設計にあたっては，欠かすことのできないコンピュータ・ツールであるとの認識をもたれてきている。その観点から本書は，CAE という有用なコンピュータ・ツールを多くの方々が実務的に役立つものとして学ぶことを留意して，実践的活用事例に重点をおいて解説を行っている。執筆者も，各分野において意欲的に CAE 技術を展開してきた著名な先進的 CAE の実践担当者が中心である。

CAE 技術を実務的観点からまとめるとともに，CAE を構成する個別技術についても基礎的事項の要約と体系化を図り，初心者でも基礎的事項から全体の把握まで十分理解できるように解説されている。

本書は 6 章からなり，各章は次の通り。◎ CAE 入門◎ CAE の個別技術◎個別技術者の適用例◎実践的活用事例◎ CAE 技術の動向◎高度 CAE 技術―非線形問題―

COMPUTER GRAPHICS WORLD

Vol. 11, No. 6　Jun. 1988
◇特集：現在のソリッド・モデリング
◇ソリッドとサーフィス
◇ソリッド・モデリングをめぐる論議
◇簡単にソリッド・モデルを生成する方法
◇ CD-ROM の未来
◇グラフィック・スーパー・コンピュータ
◇木星行き無人探査船ガリレオ号の航路をコンピュータ・グラフィックスで描く

発行所　PennWell Publishing Company
問合せ先　インターコンチネンタル・マーケッティング㈱　☎ 03(661)8373

The Visual Computer

Vol.4, No. 1　Jun. 1988
◇画家の道具としてのコンピュータの限界
◇コンピュータによって生成されたイメージと従来の版画
◇コンピュータ：創造的な魂を篭から解き放つもの
◇サイバネティックス・ジュエリー：情報化時代の装飾品
◇コンピュータ：ビジュアル・アートのための新しい道具
◇モナ・リザの検証：コンピュータ解析により得られた証拠

問合せ先　Springer-Verlag Berlin Heidelberg Tokyo, Hongo 3-chome, Bunkyo-ku, Tokyo 113, Japan

特集3　3次元機械設計用

CAD/CAMを効果的に利用するための重要なポイントの一つとして，汎用的につくられているCAD/CAMシステムを実際に使用するユーザーの仕様にきめ細かく対応するカスタム化（カスタマイゼーション）があげられる。

例えば，建築とか機械設計と一口にいっても，使用する企業によって使い方がまちまちである。PIXEL編集部が行ったいろいろなCAD/CAMユーザー調査においても，今後CAD/CAMソフトウエアを自社開発するという傾向は大幅に少なくなってきている。これまでは自前のカスタム・ソフトウエアをつくってきたが，これからは良質なパッケージ・ソフトウエアがあればそれを使いたいというのが平均的な意見である。

しかし，CAD/CAMパッケージをつくり，定期的にメインテナンスしていくため，平均的な要求仕様に応じてソフトウエアを設計しなければいけない。各企業の設計行為を市販されているCAD/CAMに合わせることは困難であるから，いかに簡単にカスタム化できるようにCAD/CAMソフトウエアが柔軟につくられているかがポイントになる。

このようなフレキシビリティに富んだパッケージ・ソフトウエアの構築というのは，なにもCAD/CAMに限定されたことではなく，どのようなアプリケーション・ソフトウエアにも適用されることである。CAD/CAM分野では，このカスタム化が他のアプリケーション以上にCAD/CAMシステム運用の大きな鍵になっていることは事実である。

これまでのようにCADとはいいながらも製図機能の領域を出なかった時代から，設計行為または製図の周辺にまで食い込んでくるようになるとこのカスタム化が一段と重要になってくる。極端にいえば，標準的に提供されているパッケージ・ソフトウエアだけでは不十分とまでいえるかもしれない。

CAD/CAMソフトウエアのカタログの「このソフトウエアはカスタム化が容易です。カスタム化の機能が充実しています」というキャッチフレーズは，おおいに歓迎したいものである。

CAD/CAM

最新技術動向とユーザーにおける使い方

Auto CADの設計思想 ―オープン・アーキテクチャ

汎用

オートデスク

CAD/CAMシステムが提供しているカスタム化の機能

AutoCADは現在，全世界で12万セットを超え，建築・機械・金型・電気などあらゆる分野で利用されている。ここまでAutoCADが設計分野に進出したのは，汎用CADとしての豊富な作図・編集機能の他に，ユーザーが用途に合わせて独自のシステムを構築するためのカスタマイズ機能もまた豊富に用意されているためである。その一つとしてAutoLispがある。AutoLispはCommonLISPに最も近いLISP言語で，AutoCADとのインタフェースをとるために，図形情報を取り扱うためのさまざまな命令をもっている。AutoLispでプログラムを作成することにより，ユーザー独自のコマンドをAutoCADに追加することが可能となる。また，AutoCADの一連のコマンドをテキスト・ファイルに書いておき，書き込まれたコマンドをすべてバッチ処理させることができる。これにより，用紙サイズの設定など作図するうえで必要な一連の環境設定をすべてマクロ化することができる。また，どのような分野でもよく使用する図形(シンボル)があるが，AutoCADはこれらのシンボルに部品番号，規格，コストなど，その図形がもつ属性をつけて登録することができる。ここで作成された属性データをファイルに書き出し，他のソフトウエアに引き渡すことにより，部品表や見積表などの作成が可能となる。

また，コマンドを実行する際に使用するメニューとして，AutoCADはスクリーン・メニュー，タブレット・メニュー，ボタン・メニュー，プルダウン・メニュー，アイコン・メニューなどを用意しており，必要に応じて各メニューから自由にコマンドを実行できるが，これらはすべてユーザー独自のメニュー

に作成することができる。前述したマクロ機能を使って，1つのメニューを選ぶことにより一連のコマンド群を自動実行させたり，シンボル名あるいはそのシンボルの形状をメニューとして表すことができる。その他にもAutoCADは，「DXFファイル」というASCII形式のファイルを用意しており，AutoCADの図面データと他のソフトウエアで作成したデータを双方向でやり取りすることができる。これにより大型CADとのデータ交換やNCテープへのデータ転送，構造解析へと発展させることができる。このように，AutoCADをベースにしてユーザーが手を加えることにより，自社に合った独自の専用CADシステムを構築できるという「オープン・アーキテクチャ」を，AutoCADは設計思想としている。

カスタム化の具体的な実施例

(1) 名称：GAUDI-SUPER2

(2) 目的：建築意匠設計用CAD

(3) 開発元：兼松エレクトロニクス

(4) カスタム構成

- スクリーン・メニュー
- タブレット・メニュー(3枚)
- AutoLispにより記述されたコマンド群
- 建築用シンボルデータ

(5) 開発経緯

当社は，従来より汎用ミニコンとCADをベースにした建築専用CADを販売してきたが，3年前にパソコンCADとして機能が高くかつカスタム化の容易なAutoCADをベースにミニコン建築CAD相当のGAUDIを開発した。その後，AutoCADのバージョンアップとともに3回のバージョンアップを重ね，名実ともにパソコン建築CADのトップの座を獲得するにいたった。

(6) カスタム化内容

- メニュー

GAUDIのメニューは，スクリーン・メニューとタブレット・メニューの2つに分かれ，スクリーン・メニューは階層選択方法により使用頻度の低いもの，例えば図面縮尺設定などが配置されている。タブレット・メニューは，直接選択方法により使用頻度の高い柱・壁・開口などのコマンドが配置されている。

- AutoLispによるプログラミング

GAUDIのコマンド群はすべてAutoLispによりプログラミングされており，他の言語をあえて使用していない。これによりAutoCADが稼働する多数のコンピュータへの移植性のあるアプリケーションを実現している。

カスタム化に対する方針

前述のようにAutoCADは，その開発当初より「カスタム化」の必要性を考えて対応してきた。そしてアプリケーション・ソフトとして多くのサードパーティより開発，販売が行われている。

現在，世界で400以上のソフトウエアが，また日本では120以上のソフトウエアがオートデスク社に登録されている。アプリケーションの分野も，建築，機械，電気・電子など幅広く，使いやすいシステムとして提供されている。

今後は，現在と同様，より使いやすいシステムとなるようにするとともに，マン・マシン・インタフェースの向上など基本となるべき土台を固める。また，3次元機能の充実，シェーディング，アニメーションなど先進的な技術にも取り組み，多くの方が手軽に使うことのできるシステムを提供していく予定である。

問合せ先：オートデスク㈱　営業部

〒150　東京都渋谷区恵比寿1-19-15

☎03(473)9511

開発元：Autodesk Inc.(米国)

製品のフルカスタム化を基本とするCADIXシステム

キャディックス

CAD/CAMシステムが提供しているカスタム化の機能

CADIX社におけるCAD/CAMシステム開発の基本的考え方は，作り出される製品は同じであってもユーザーにはユーザー独自の業務/設計フローがあり，おのずからCAD/CAMシステムに対する要求もユーザーごとに異なる。この要求を満足させるために「フルカスタム化が可能なシステムの開発」があった。このためハードウエアおよびソフトウエアとも，ベースそのものからフルカスタム化が十分可能なシステムとなっている。

カスタム化のツールは以下の通りである。

(1) ハードウエア

- 自動図面読取り装置：ユーザーの取り扱っている既存図面などのチェックやデータベース化のために，ユーザーの仕様に合わせてシステムへ自動的に取り込む。
- 高速汎用EWS：ユーザーの各種ソフトウエアに対する要求を満足させるために多くの既存ソフトウエアが利用できる。
- 各種周辺装置：ユーザーごとのシステム構成に合わせた周辺装置のサポート。

(2) ソフトウエア

- 汎用グラフィック・データベース：256レイヤー，256サブレイヤー，256階層構造，各種図形エレメントなどのベースをもち，ユーザーごとに自由に設定可能。
- ユーザー・インタフェース：ユーザーごとに作成可能なグラフィック・メニュー。
- ユーザー言語と業界標準フォーマットのサポート：インタラクティブにグラフィックにアクセス可能なユーザー言語と，IGES, GERBER, STREAMなどの業界標準フォーマットや各種中間フォーマットに対するFORTRAN, Pascal, Cなど

の汎用言語によるバッチ的アクセス。

●豊富なソフトウエア資源：過去長年にわたってフルカスタム化を前提として行ってきたマップ，メカニカル，PCBなどの各種アプリケーションに対する開発の結果としての多くのプログラム群とそのノウハウ。

カスタム化の具体的な実施例

電子・電気設計分野において最も多く存在する運用形態として，設計者が回路設計を行いアートワーク展開を外注業者に依頼し，出来上がった手張りやフィルム・アートワークと回路図のチェックを設計者自身が行うというものである。この形態の問題点としては，

●チェックに設計者自身が多くの時間を費やしている

●既存CADシステムへの取込みに時間がかかるため，設計変更に素早く対応できない

があげられる。これらの問題点を解決するために，このユーザー向けにフルカスタム・システムを開発，納入した。

システム構成は，自動図面読取り装置と高速EWSから成り立っており，これらはネットワークで既存CADシステムとつながっている。この回路図とアートワークのチェックの合理化のためにアートワークを自動図面読取り装置でベクトル化し，パターン・データとドリルホール・データを認識してトレースリスト・データ(同一パターン内にあるドリルホールの同電位情報)を既存CADシステムへオンライン転送する。回路図データと部品配置データ，トレースリスト・データがCAD上で自動照合されるため，回路図面とアートワークとの目視チェックに比べて大幅な時間短縮を実現した。また，このアートワークは，CADデータベースとして従来の手入力に比べはるかに高速に取り込んでおけるので再度外注業者へ依頼する必要がなく，CAM情報といった製造情報もすぐ出力できるためにますます増えてくる設計変更に対して素早い対応が可能である。

カスタム化に対する方針

最初に述べたように，初期の開発の時期からCAD/CAMシステムのフルカスタム化を基本としているため，今後もこの方針で進む。また，それが可能なシステムとして十分に考慮されている。ユーザーの現行業務フローおよび要求の把握も，よりユーザーの中に入り込んだかたちで行い，ソフトウエアはもちろんのことハードウエアも含めた開発，納入を行い，納入後も，よりシステムの完成度を上げるためにバージョンアップを行っている。

CADIXシステムは，これまで自動図面読取り装置から出発し，電子・電気設計分野，機械設計・設備設計分野，地図応用分野，そしてアパレル分野とユーザーの要求に合わせたフルカスタム化を目指して発展してきたが，今後もユーザーの要求は，

●さらに高度な認識技術を利用した本来の意味での自動読取り

●各ユーザーの要求に合わせたシステムのインテグレート

●衛星通信などを利用した各種データの通信(ネットワークの広域化)

へと進んできており，これらのニーズに対し，今後もこれまでの経験と高度なハードウエアおよびソフトウエア技術を核にし，「各分野ごとに特化した認識技術の拡張システム間の有機的結合性をより密にする開発」をテーマとして進んでいくことになるであろう。

問合せ先：㈱キャディックス　営業本部
〒154　東京都世田谷区桜新町1-12-10
☎03(427)0401

2次元汎用CAD/CAMシステムTekniCAD/CAMのカスタム化

ソニー・テクトロニクス

CAD/CAMシステムが提供しているカスタム化の機能

(1)　TekniCAD

TekniCADは汎用の2次元CADシステムである。したがってカスタム化のニーズは数年前よりあり，これに対応するためにマクロ言語を開発し，機能の充実を図ってきた。マクロ言語はCAD用のBASIC言語ともいえ，TekniCADの操作を覚えれば簡単にプログラムすることができる。

● TekniCADがもつメニューの各機能をマクロ言語で使用することができる。マクロ言語はTekniCADのコマンド入力モードで用いる3文字のニモニックを利用しており，覚えやすい。複合した一連の操作を1回のオペレーションで実行できる。

● TekniCADのデータベースへのアクセスが可能である。データベース構造を意識せずに図形情報や属性データの検索，追加，削除が可能である。

● TekniCADの一つに，オペレーション中にOSのモードへ割り込む機能がある(ビジットOS，**図1**)。この機能を利用すると，CやFORTRAN言語により作成されたプログラムをマクロ・プログラムの中で実行させることができる。ファイルを介してユーザー・アプリケーション・プログラムとマクロ・プログラム間のデータ授受も可能である。

●図面データのピック，座標値の入力，システム環境パラメータの更新，各メッセージエリアへの文字出力など，きめ細かい図形処理が可能である。

●マクロ自動生成機能をもつ。入力された一連の操作を自動的に記憶できる。不要な操作が入った場合，ステップモードにより操作を確認後，エディタで修正する。

その他，シンボル・データの利用によるカスタム化も可能である。数多くのシンボルを登録することにより，専用化したレイアウト・システムができる。

(2)　TekniCAM

TekniCADにより作成された図面から2.5軸までのNCデ

ータを作成する。マシニング・センター/フライス，ワイヤーカット，ガスレーザー，タレット・パンチをサポートする。NC マシンは，制御装置との組合せにより異なるコードを必要とする場合が多く，これに対応するためにポスト・プロセッサを作成した。ポスト・プロセッサは工具，切削条件，加工工程(**図 2**)，ポスト・パラメータなどのファイルをもち，ユーザーが各ファイルに固有のデータを入力できる。これらを登録することにより，各種 NC マシンに対応することが可能となり，ユーザーはポスト・パラメータの名称を入力するだけで異なる NC データを作成できる。

カスタム化の具体的な実施例

(1) TekniCAD

■マクロの利用

- TekniCAD で作成された 2 次元データから 3 次元データを作成する機能を実現している。ワイヤー・フレーム・モデルから隠線消去したり，カラー・シェーディングする機能まで付加している。メニューも完全に独自の形式を実現している。また，3 次元のプリミティブを定義し，CSG 方式による 3 次元入力も可能としている。アイソメ図の作成に有効である。
- 自由曲線の入力を実現した例もある。Bézier 曲線，B-Spline 曲線などを入力された連続点から近似計算し，作図する。近似計算のスピードアップを図るため，計算プログラムを別途に FORTRAN で作成している。
- タイヤの断面形状を自動作成するマクロを用意したユーザーもある。諸条件を対話形式で入力した後に，断面図や寸法線，各種表への記入，図面データの格納などを自動的に行う。

■シンボルの利用

設備図面専用のシンボルを約 40,000 件入力し，図面作成，積算作業の簡単化を図っているユーザーがある。固有のシンボルを配置しさえすれば後は配線，注釈の入力だけでよく，効率の良い専用システムといえる。

(2) TekniCAM

TekniCAM 一式で 20 種以上もの工作機械をサポートしているユーザーがある。もちろん制御装置の型名/バージョン，サブ・プログラムの使用などが NC マシンにより異なるが(**図 3**)，ポスト・パラメータを変更することにより対応が可能となっている。また，同一の製品を作成する場合でも，メーカーにより加工する手順が異なる。加工工程テーブルを作成することにより，独自の手順による加工が可能となる。

カスタム化に対する方針

汎用の CAD/CAM システムであれば，カスタム化のニーズは必ずでてくる。しかし，汎用システムなのですべてのニーズに対応するのは難しい。大多数のユーザーが望まない機能も含まれてしまう可能性があるからで，CAD/CAM メーカーとしては，専用システムを開発してある決まったマーケットに販売するか，汎用システムにより広いマーケットに販売するかのどちらかであろう。専用システムは限られたことを可能にするが，多目的な利用には不向きな面がある。一方，汎用システムは多目的な利用に向いているが，専用システムに比べてきめ細かいサポートは難しい。汎用 CAD/CAM システムを販売する当社としては，ユーザーが独自にカスタム化できる使いやすいツールを拡充し，解放していきたいと考えている。基本的には，システム本体とカスタム化できる部分が切り離せる形式をとる。その一つの方法として考えられるのが，C または FORTRAN のプログラムから TekniCAD の各機能を直接コールできるよ

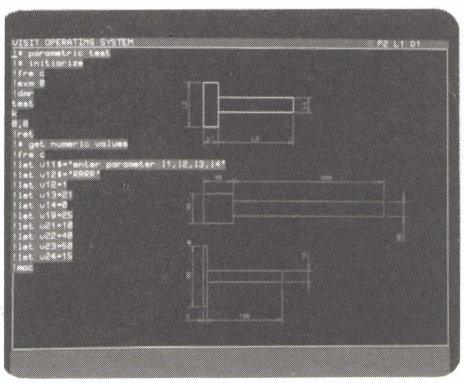


図 1　パラメトリックな作図例

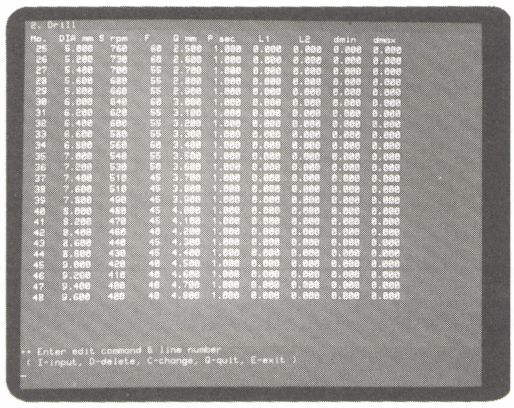


図 2　加工条件入力

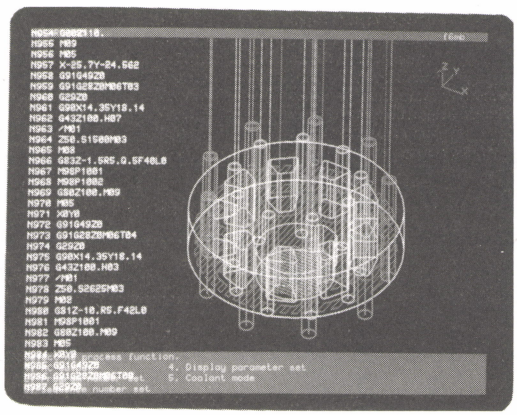


図 3　加工シミュレーション＋NC テープ出力

うにする方法である(ライブラリ化)。作成したユーザー・プログラムは，TekniCADの中からサブ・プロセスとして起動できる。この方法は，マクロ言語によるカスタム化に比べてきめ細かく大規模なカスタム化を実現できる。TekniCADをベースにした新規システムの構築も可能である。現バージョンではサポートされていないが，将来はサポートを予定している。

現在のTekniCADをベースにした別システムの構築は行わない。実際に電気系CADの構築を試みたこともあるが，結局作り直してしまった経験もある。現状では，マクロ言語とユーザー・プログラムの利用によるカスタム化を推奨している。

カスタム化のためのインプリメント作業は基本的にユーザーが行う。当社は，そのための良いツールを提供することに集中し，汎用的な機能の拡充に力を注いでいく方針である。

問合せ先：ソニー・テクトロニクス㈱　情報機器部
〒141　東京都品川区大崎1-6-4
☎03(779)7602

CADPAC-SPS/FAS/VENUSのカスタム化

デザイン・オートメーション

CAD/CAMシステムが提供しているカスタム化の機能

デザイン・オートメーションでは，ユーザーのカスタム化要求レベルに応じて多様な手段を用意している。カスタム化の深さとカスタム化を行うユーザーの技術レベルに合わせて，対話的な条件設定から高級言語による大規模アプリケーションの開発，さらにはCADシステム本体を構成するライブラリの提供まで可能である。

(1)　メニューおよびオペレーション方法のカスタム化

ユーザーが，コマンド・メニューおよびオペレーション方法の一部を自分の望むかたちに設定する機能である。すべて専用ソフトウエアにより対話的に設定可能であり，ユーザーはコンピュータやプログラミングの知識を一切必要としない。コマンド・メニューのメッセージとその位置を自由に指定でき，キーボードからの指定方法も設定可能である。タブレット・メニューの場合も同様，任意の形状でCAD機能を活用して作成できる。

(2)　システム・パラメータによる各種処理条件の設定

作図・編集・運用などに関するさまざまな条件を一括してパラメータ設定できる機能である。寸法記入の詳細条件，プロッタ出力条件，ズーム表示の倍率など，数十項目にわたる設定が可能である。設定方法は，パラメータ・ファイルをエディタにより書き換える方法から，対話的に設定する方法までユーザーレベルに応じて用意されている。

(3)　インタプリタ言語によるユーザー・アプリケーション

国内で最もポピュラーなN88-BASIC準拠の高水準インタプリタ言語により，高度なユーザー・アプリケーションを自由にかつ短期間で開発できる。BASIC言語からCAD基本部のさまざまな機能を呼び出し，高速度で処理することができる。作図，操作，ファイル入出力など，図形処理に必要なすべての機能の他，BASICの各種機能を活用できる。

(4)　コンパイラ言語によるユーザー・アプリケーション

C言語により，高度で大規模なユーザー・アプリケーションを開発できる。標準のC言語を用いて開発できるため，特殊な専用言語をマスターする必要がなく，効率的で自由度の高い開発が可能である。C言語の関数としてCAD基本部の各種機能を用いることができる。

(5)ライブラリによる完全カスタム化

C言語で記述されたライブラリを組み合わせて用い，ソースレベルでの改造も加えることにより，データ構造をも含む完全なカスタム化を行うことができる。十分な開発力を有するハイレベルなユーザー向けのカスタム化方法である。

カスタム化の具体的な実施例

インタプリタ，コンパイラ，ライブラリの3形式について，実例をあげて説明する。

(1)　インタプリタ言語によるカスタム化例

CADPAC-SPS用のユーザー言語「DA-TALK」によるユーザー・アプリケーション作成の例は無数にある。関西にある大手土木関連コンサルタントの場合は，高速道路設計のために，各種の曲線を発生するアプリケーションを「DA-TALK」を用いて短期間・小人数で開発し，実用化している。

また関東の大手電気製品メーカーでは，電気回路作図用の機能として，配線上に部品を配置した場合，その下にある線を自動的に切り取って消去するポップイン機能を実現し，多数台の使用と相まって高い実用効果を上げている。

(2)　コンパイラ言語によるカスタム化例

CADPAC-FASのユーザー・アプリケーション開発機能を用いて，C言語により業種別アプリケーションが多数作成され，使用・販売されている。見積り機能，部品集計，CAM連動などの機能をもつ配電盤設計製作支援システム，各種部品の自動配置により，あるレベルでの自動設計を可能とした電気設備設計専用CADシステム，その他，木工用CAD，点字作成のためのCAD/CAMシステム，カッティング・プロッタと組み合わせたシール作成システムなど，高いレベルのカスタム化を行った専用CADが多数存在する。

(3)　ライブラリによる完全カスタム化の例

C言語ライブラリの全ソースを用いることにより，OS-9やUNIXなどの別OSへの移植，68000系などの別CPUへ移植した例がある。さらに，データベース構造を改変し，ユーザーの望む属性データベースを付加した例がある。また，必要最小限の軽いシステムへと縮小させたケースもある。

カスタム化に対する方針

今後とも，カスタム化の度合いと，ユーザーレベルのすべて

の階層に幅広くこたえられるカスタム化手段の品ぞろえを強化していく予定である。

(1) 浅いレベルのカスタム化をより容易にする

オペレーションのカスタム化や既存機能の一部仕様設定など浅いレベルでのカスタム化をより容易に，プログラムを意識することなく可能となる方法を強化拡大する。

(2) マクロコマンド作成をより容易にする

既存のコマンドを組み合わせ，新たなコマンドとするマクロコマンド機能を充実させ，プログラミングの知識を要せずに対話的に新しい機能を作成できるようにする。

(3) 基本関数またはサブルーチンの高級化

インタプリタおよびコンパイラ言語からコールされる関数またはサブルーチンのレベルをより高いものとし，高度なアプリケーションを少ないステップ数でコンパクトに実現可能とする。

(4) アプリケーション領域を拡張する

インタプリタ，コンパイラとも，そのプログラム領域を拡張し，より複雑で大規模なアプリケーション作成を可能とする。

(5) ライブラリの整備とカスタム化サポート力の強化

大幅な変更を必要とするハイレベルのカスタム化を支援するため，ライブラリの機能をより充実させるとともにカスタム化そのものを支援し，あるいは受注開発し得る組織を用意し，CADのカスタム化全体を強力にかつ効率良くサポートできる体制を整備する。

問合せ先：デザイン・オートメーション㈱　営業部
〒181　東京都三鷹市上連雀 1-1-5
☎ 0422(55)5731

CATISのカスタム化

東洋情報システム

CAD/CAMシステムが提供しているカスタム化の機能

CATIS/DRAWには以下のカスタム化ツールがあり，目的によって使い分けることで効率的に専用CADを構築することができる。

- ●カスタム・メニューシート
- ●ユーザー・コマンド機能（マクロ化機能）
- ●パラメトリック図
- ● UDI(User Data Interface)
- ● UCSP(User Control Subroutine Package)

(1) カスタム・メニューシート

CATISの標準メニューシートの中の必要なコマンドを抜粋し，ユーザー・コマンド，部品，文字などと混合してレイアウトし，ユーザー独自のメニューシートを作成する機能である。フロー型メニューシートや業務別の専用メニューなど，作業上に最適なメニューを作成できる。

(2) ユーザー・コマンド

CATIS/DRAWのコマンドを組み合わせることにより，ユーザー専用コマンドを作成する機能である。ユーザー・コマンドは，変数・演算式・条件判断文・ユーザーとの対話処理・ファイルとの入出力なども組み込むことができ，より効果的な機能を実現することができる。作成方法には，既存入力図面からの自動生成や操作手順を蓄える方法もある。

(3) パラメトリック図

図形間の距離・角度・位置などをあらかじめ変数として定義し，この変数に値を設定することにより類似形状の設計を迅速に行う機能である。変数の定義は管面を見ながら対話型で行える。また，値の設定は個々の変数に値を与える方法と，テーブル名の指定による一括入力が可能である。

(4) UDI(User Data Interface)

CATISデータベースへアクセスするためのツールである。ディスプレイ画面での確認を必要としない定型的なバッチ型自動製図，あるいは他システムとのデータの共有化などを行うインタフェースである。

(5) UCSP(User Control Subroutine Package)

ユーザーのプログラムをCATIS/DRAWに組み込むためのツールである。CATIS実行中にインタラクティブ・モードでリアルタイムに処理できるので，設計部門で蓄積されたノウハウを組み込んだ独自のシステム構築が可能となる。

カスタム化の具体的な実施例

CATISを利用したカスタム化の例のうち主なものについて記述する。

- ● K鉄工社：ダクト積算システム
- ● A社：金型標準メニュー，図面管理(検図・検印・検索)
- ● Tガス社：建物内配管自動設計システム
- ● B社：環境機器レイアウト設計，見積りシステム
- ● C社：アルミサッシ収まり図設計システム
- ● D社：自動車生産ライン治具設計システム
- ● E社：図面管理(光ファイル)，骨組解析・設計・積算

カスタム化に対する方針

以前はCAD/CAMシステムを構築するということは，ユーザーがコンピュータを利用して独自にシステムを設計・開発することであった。しかし最近では，手っ取り早く既存の市販システムを選択・導入することを意味する場合が多い。問題はここから発生する。

設計・製造の業務は部門ごとにその要求が多彩であり，パッケージ・ソフトウエアの世界で汎用的にこれらの業務を支援するのは困難である。

既存の市販システムを導入する場合，上記の問題点を十分考慮したうえで，いかに「自社システム」として構築・運用できるかを検討しなければならない。CATISは，その意味で「自社

システム(カスタム化)」とするためのさまざまな方法がツールとして用意されている。

また，CATISをツールとしてユーザー独自のシステムを構築する場合，その開発を請け負うことも可能である。

問合せ先：㈱東洋情報システム　応用システム営業部
〒105　東京都港区新橋1-11-7
☎03(575)4021

UNICADのカスタム化

日本ユニシス

CAD/CAMシステムが提供しているカスタム化の機能

UNICADは500以上もの豊富な基本コマンド群を背景に以下に示すカスタム化機能により，設計・製図から統合設計生産システムまで幅広い分野で適用されている。

(1)　UDL(Unicad Design Language)

UDLは手続き型プログラミング言語で次のような特徴をもつ。

- ●図形処理機能(図形型の変数，POINT型の演算子／関数・図形入出力)
- ● DB操作機能(検索，削除，変更，創成)
- ●コマンド実行機能(UNICADコマンドのほとんどすべてを呼び出すことができる。DB検索結果，図形処理結果を引数として与えることができる。実行結果の図形データやDB識別子をプログラム内で使用できる)
- ●外部入出力機能(日本語を含む文字列の入出力。画面上の任意の位置にテンポラリ図形・文字の表示ができる。FORTRANファイル入出力。テキスト入出力。テーブル入力)
- ●バッチ処理機能

(2)　メニュー再配置

コマンド・メニューをタブレットに自由に配置できる。UDLで作成したプログラムをメニューに配置することができる。

(3)　UFIO(Unicad File Input Output Utility)

DBとFORTRANファイルとの相互データ変換ユーティリティ。UFIOで変換されたFORTRANファイルを介して他システムとの連動ができる。

(4)　結合RAN

FORTRANなどで作成したユーザー・プログラムをコマンドとして実行することができる。

(5)　漢字メニューの再配置・辞書登録

外字，文節，マークなどの登録ができる。漢字タブレット・メニューの配置が自由にできる。またカナ漢字変換機能で使用する辞書の登録ができる。

(6)　寸法モーダル値の設定

寸法線丸め幅，寸法形状などのモーダル値の初期値を設定することができる。

カスタム化の具体的な実施例

図1はUFIOと結合RANを利用して，社内製図規格に合わせた図面をオンラインでプロッタに出力するという処理の概要である。

メニュー再配置による操作性向上から，自動製図や設計手順の取込み，さらには**図2**に示すような他システムとの連動化まで，UNICADのカスタム化の機能はその要求に応じて幅広く柔軟に対応することができる。

カスタム化に対する方針

ユーザー独自のシステムを構築するうえでUDLはカスタム化の強力な手段であり，ユーザーの個々の要求に対応できる能力がある。またユーザーは，UFIOなどデータベース・インタフェースを利用し，生産管理，情報検索，一般事務処理などのソフトウエアとUNICADを連動させることもできる。

UNICADは，当社が提供している豊富な商品群と併せて設計から生産までの幅広い領域で利用される統合設計生産システムを指向しており，今日までユーザー内での利用形態や適用業務の拡大を考慮しながら各機能の充実が図られている。

また，万全な開発体制およびサービス体制により，教育，導

◉図2は次ページ。

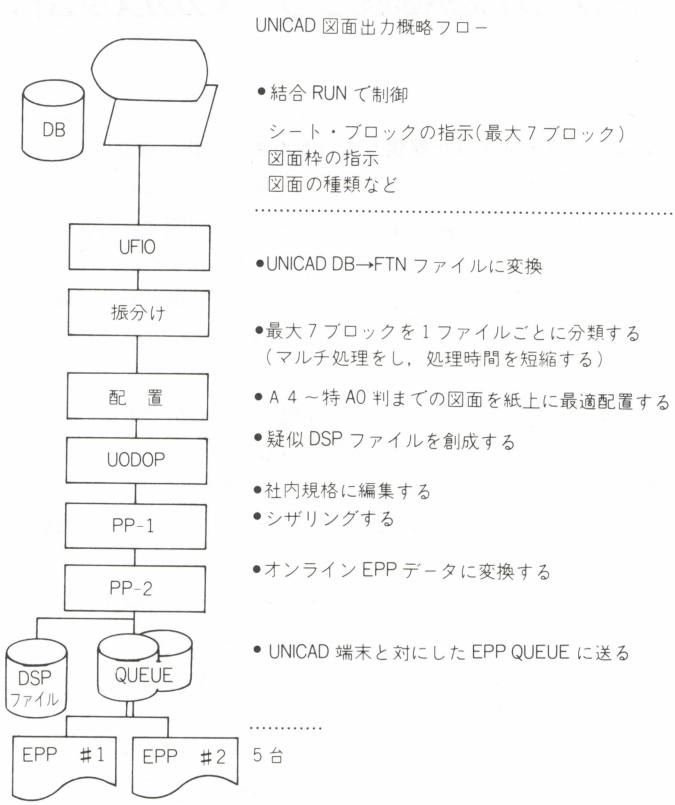


図1　A社における図面出力処理概要

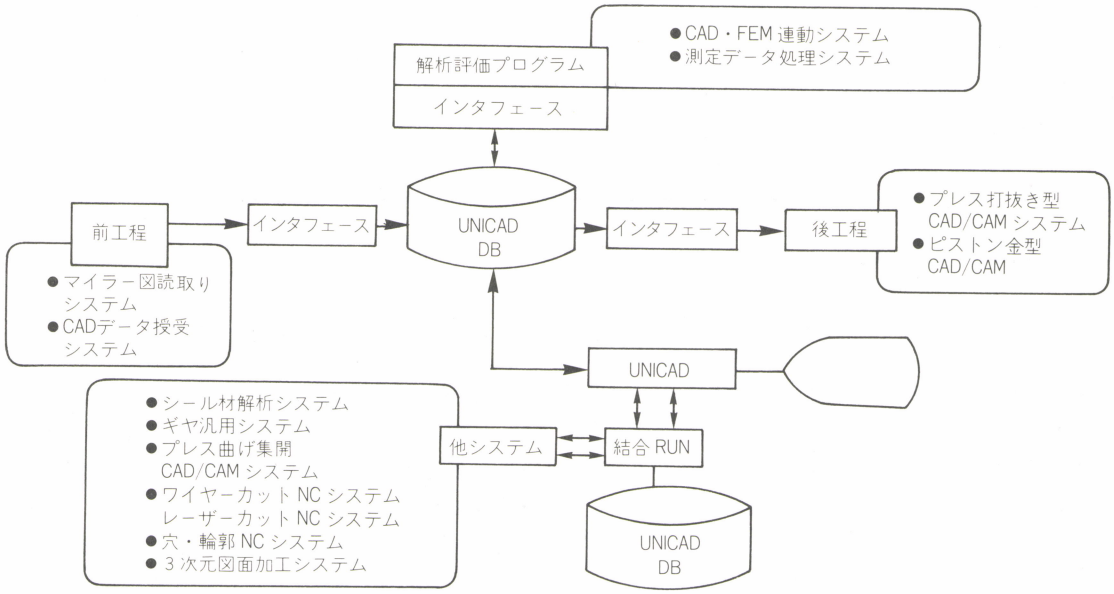


図2　A社における UNICAD 連動システム

入，適用，カスタム化などきめ細かな対応を行っている。

今後ともこの姿勢を堅持し，新技術の取込みなども積極的に行っていく方針である。

問合せ先：日本ユニシス㈱　CIM マーケティング部
〒107　東京都港区赤坂 2-17-51
☎ 03(587)8692

SOLUTION3000 シリーズのカスタム化

ファモティク

CAD/CAM システムが提供しているカスタム化の機能

SOLUTION 3000 のオープン・アーキテクチャ設計は，ユーザーに操作性の向上と専用アプリケーションの開発を可能にした。特にカスタマイズするために，ソースレベルやサブルーチン形式の変更といった専門的なプログラムの知識がない一般のユーザーでも簡単にカスタマイズできるように設計されている。

ユーザー・メニューはいくつも作成することができる。しかもすべてに専用エディタが付属している。

●スクリーン・メニュー(ポップアップ／プルダウン)
プルダウン・メニューはよく使うコマンド群の見出しを常時，スクリーン上に表示できる。
ポップアップ・メニューはマウスやタブレットの第2ボタンに割り付けられており，それぞれ画面の左右に別々のメニューをもつことができる。どちらのメニューを表示させるかはボタンを押した位置により決定される。

●タブレット・メニュー
よく使うコマンドはタブレット上に登録できる。

●ショートキー機能
すべてのキーにマクロを割り付けることができる。
この機能によってコマンドを全部入力する必要はない。

●強力なマクロ機能
1つのファイルには最大 50 マクロまで登録可能である。
また1つのマクロファイルには最大5層までネスティングすることができる。さらにマクロ作成用エディタには文法チェック機能も付加されているために，メインプログラムに戻りデバッグする必要がない。

●条件分岐やループ，ファイル管理，変数設定，演算といった機能をもつ BASIC に似た覚えやすいアプリケーション開発用言語「DGL」を装備している。

●アプリケーション実行のために利用できるメモリサイズには制限がない。したがって，どのようなアプリケーションもメモリサイズを気にすることなく開発することが可能である。

●他の CAD システムとのデータ交換には DXF ファイルや IGES，IGES サーフィスなどを用意している。

●3次元データを NC 用のデータとして利用するための APT ファイルは ASCII 形式で(x, y, z および i, j, k)の座標値を出力することができる。

●マルチ・ポスト・フォーマッタ言語によるユーザー専用の加工機械に NC データを送ることができる。

カスタム化の具体的な実施例

標準で設定されているメニュー例を以下にあげる。

●スクリーン・メニュー
プルダウン・メニューにはCAD／サーフィス／NCなどのモジュール別のメインメニューがロードできる。
左側のポップアップ・メニューには，ズームやパンなどの画面操作に関するコマンド，右側にはXY／XZ／YZ／ISOなどの視点を設定するコマンドが割り付けられている。もちろん，このようなコマンドは他のコマンド実行中にも割込みが可能である。
●タブレット・メニュー
作図を中心としたコマンドが登録されている。
●ショートキー機能
作図に関するコマンドを中心にしたマクロがすべてのキーに割り付けられている。
●マクロ機能
いくつかの便利なマクロコマンドが登録されている。
●他のCADシステムとのデータ交換にはDXFファイルやIGES，IGESサーフィスなどが用意されている。
また各自動車メーカー(GM，フォード，クライスラー)向けの標準フォーマットとのコンバータにより，実際にデータをやり取りしている。
● APTファイルにより作成した3次元データを3軸から5軸までのNCで利用できる。もちろん，ASCII形式で記述されたこのファイルを他のエディタやアプリケーションなどで編集し，再利用することもできる。
●マルチ・ポスト・フォーマッタ言語はユーザーの工作機械用コードを生成できる。

カスタム化に対する方針

●誰でも利用できるユーザー・インタフェース
SOLUTION3000シリーズでは3次元CAD，サーフィス，NCなどすべてのモジュールにおいてソースまたはサブルーチンレベルによるカスタマイズというようなものではなく，プログラムの知識がない一般のユーザーにも簡単にカスタマイズできるようなオープン・アーキテクチャ設計をとっている。
●専用エディタを装備
カスタマイズをより簡単に行うことができるように，カスタマイズのための専用のエディタも別に用意されている。マクロ用エディタには文法チェック機能も付加されているため，メインプログラムに戻ってデバッグする無駄を省いている。
●アプリケーションに利用できるメモリサイズ
アプリケーションを作成する場合に，メモリサイズを気にする必要はない。
●データの一貫性
SOLUTION3000シリーズではCADモジュールやサーフィスで作成されたデータが，そのまま3次元のNCデータとして利用できる一貫性をもっているため，別々に開発されたCADシステムやCAMシステムのようにデータのやり取りに別の変換プログラムを用意する必要がない。
● APTファイル
ASCII形式で記述されたAPTファイルにより，作成した3次元データ(x, y, zおよびi, j, k)を3軸から5軸までのNCで利用できるようにしてある。
●データ・コンバータ
他のシステムとの間でDXFファイルやIGES，IGESサーフィスによって図形データの共有が可能である。

問合せ先：ファモティク㈱
〒150　東京都渋谷区代官山7-8
☎03(780)4681
開発元：Micro Engineering Solutions,Inc.(米国)

ICADのカスタマイズ機能とその考え方

富士通

CAD/CAMシステムが提供しているカスタム化の機能

ICADは，ユーザー固有のCAD/CAMシステムの基礎となることを最大のねらいとして，当社独自で開発したシステムである。

ICADのシステム体系を**図1**に示す。ICADは標準的な設計・製図業務およびカスタム化を支援するICAD/SDS(Standard Design System)と，業務ごとの設計機能を担当する業種パッケージ群からなる。このうちカスタム化に最も関係するのがICAD/SDSであり，数々の開発支援機能をもっている。

● ICAD/SDSの開発支援機能

ICAD/SDSは大きく基本プログラムとコマンドに分かれる。

基本プログラムの中心は，オペレータとの対話を行ってCAD機能の実行に必要な入力パラメータを取りそろえる対話部と，図形データベースを管理し端末への図形表示も行う図形処理部である。コマンドは線を引く，部品を配置するといった個々のCAD機能を実現している部分であり，一般の設計・製図に必要な標準のものがあらかじめ用意されている。ユーザーはコマンドの位置に自分のプログラムを組み込める。以下，プログラム組込みを容易にするための機能を説明する。

(1)　プログラム組込みインタフェース

ユーザー・プログラムはFORTRAN77で作成され，CALL文で図形処理部を呼び出して機能の実行を命令する。このため各種レベルのインタフェースが200以上用意されている。

一般に対話型図形処理プログラムのステップ数の6〜7割は図形の入出力部分だといわれている。ICADでは対話部と図形処理部の工夫により，ユーザー・プログラムの負荷を大幅に軽減している。インタフェースの特徴は次の通りである。

●ユーザー・プログラムは業務処理に専念できる　●インタフェースのレベルが多様　●バッチプログラムの作成も容易

(2)　データベース

(図1は117ページ。)

材料集計，NCデータの自動生成，解析プログラムのためのデータ作成といった業務をユーザー・プログラムとして作成するときには，単なる絵のデータだけでなくデータとデータの関係や非図形データが必要になってくる(業務データ)。ICADではこのような業務データを次の4つのレベルで扱える。

●業務特有のセグメントの作成　●任意の情報を属性として付与　●図形と図形の間の接続情報の管理　●汎用データベースとの連動

(3)　文字データのための対話機能

業務に依存したユーザー・プログラムでは，業務固有の文字・数値データを対話で扱う必要がでてくる。

ICADには日本語フルスクリーン画面のフォーマットが簡単に定義できるツールがあり，ユーザー・プログラムは可変データだけを意識すればよいようになっている。

(4)　標準コマンドの利用

ICADの標準コマンドを部品として利用し，ユーザー・プログラムの開発量を減らすことができる。

(5)　定型処理への対処

ある一種のユーザー・プログラムや標準コマンドを連続して実行したい場合の対処として2つの手段を用意している。

一つはマクロ機能であり，コマンド列をファイルに記述しておくことにより，対話オペレーション時にいつでも呼び出せる。

もう一つは，対話のために作ったプログラムをそのままバッチプログラムとして動かす機能である。

カスタム化の具体的な実施例

●適用事例

ICADをカスタム化して使用中の適用業務を**表1**にまとめた。この表では製図だけに使っている例やICADの業務パッケージをそのまま利用している例は除いてある。カスタム化の代表例として，大和ハウス工業における住宅設計，東洋エンジニアリングにおけるプラント設計，出光興産における設備管理，古河電気工業における電気設計，若築建設における土木設計などがある。

また当社もICADのカスタム化機能をフルに活用している。プリント板設計システムをはじめ板金加工システムや順送り金型設計システムは，当社の工場部門で開発されたものであり，それをベースに商品化したものである。

カスタム化に対する方針

ICADのカスタム化に対する考え方を示す。

(1)　基本部と応用部の分離

ICAD全体の設計方針は，基本部(SDS)と応用部(ユーザー・プログラム，業種パッケージ)の完全分離である。

企業の基幹システムたり得る信頼性の保証，レベルアップ時のユーザー資産の保証(互換性)などは，基本部で解決されている。

ICADではこのような技術を盛り込んだ基本部に開発支援機能を付加し，応用部を作り込める(カスタム化)ように配慮されている。

(2)　データベースの重視

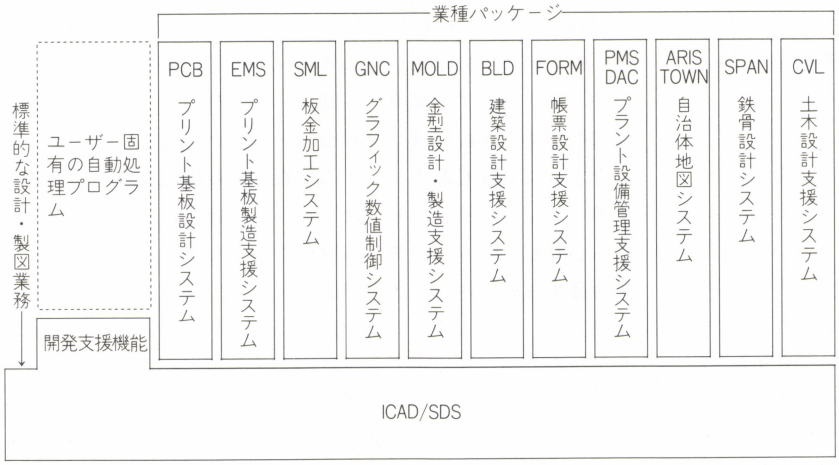
ICADが提供する図形データベースでは，応用部を作るうえで必要となる非図形データ(設計ルール，部品特性など)を入れ込める柔軟性とそれを効率良く扱う仕組みが配慮されている。

(3)　総合的なプログラム開発効率の向上

ICADでは，単にプログラム開発のためのインタフェースが公開されているというレベルではなく，プログラム開発のさまざまなフェーズを効率的に行う仕組みを提供している。これらの中には開発を行う人のための教育(機能と導入，コマンド開発，自習システムなど)や，CAD専任SEの全国配置によるSE

表1　カスタム化の適用業務例

機械	一般機械	工作機 バルブ 軸受 ベルトコンベア モーター 空調機 現像機 ボイラー ポンプ 金庫
	精密機器	レンズ レンズ金物 時計部品
	金型	絞り金型 順送金型 鋳造金型 粉末冶金金型
電気	配電盤シーケンス回路 ワイヤー・ハーネス 照明回路	
建設	建築	企画・意匠・設備 施工（矩体，鉄骨） 見積・積算
	住宅	プレハブ住宅
	建材	エレベータ サッシ シャッター パーティション 厨房機器
	エンジニアリング	プラント配管 設備配管
	鉄骨構造物	鉄塔 仮設構造物
	土木	橋梁下部構造 土地造成 ダム 港湾 水門 河川・護岸
地図作成		
設備管理	石油プラント 原子力プラント 送電信線	
ソフトウエアCAD		
サービス	ダイヤグラム	



◀ 図1　ICADのシステム体系

サポートも含まれる。

最近のICAD利用者の動向をみると，ユーザーがカスタム化を行うだけにとどまらず，ソフトハウスが開発したICADベースのパッケージをユーザーとして利用したり，数社が共同でプログラム開発を行ったり，開発したプログラムをユーザー同士で流通させたりという多様なかたちがでてきている。

ICADベースのパッケージやコマンドの流通に積極的に取り組み，「その企業にとって最も役立つソフトウエアを手に入れること(カスタム化)」をより容易にしていく所存である。

問合せ先：富士通㈱　メカトロニクスシステム部
〒105　東京都港区西新橋3-21-8
☎03(437)5111

インターグラフ・システムのカスタム化の機能と実施例

武藤工業

CAD/CAMシステムが提供しているカスタム化の機能

●ユーザー・コマンド言語

BASICに似た習得が容易なプログラミング言語。基本コマンドのマクロ化や新機能の作成に用いる。条件分岐やサブルーチンコールなどの他に，スクリーン・メニューのコントロール，高級言語プログラムとのインタフェース，データベース・マネージメント・システムとのインタフェース機能を備えている。図形データファイルへの直接アクセスや図形処理ソフトの実行時パラメータのコントロール機能により高度な機能の実現も可能。インタプリタ型だが実行速度は速い。EWSやパーソナルコンピュータではコンパイルしてスピードをさらに上げることもできる。

●プログラム・インタフェース

図形処理ソフトと高級言語(FORTRAN，C)で作成するプログラムを連動させるための機能。インタフェース・ルーチンのライブラリがあり，それを高級言語プログラムから呼び出して使う。図形ソフトの基本コマンドはすべて使用可能。ユーザー・コマンド言語やデータベース・マネージメント・システムとのインタフェースの他に基本コマンドを拡張した機能ももつ。高度な機能の開発やユーザー独自の大規模なアプリケーションの開発に用いる。

●メニュー・システム

さまざまな形式のメニューがあり，用途や状況に応じて適切なものを使い分けることができる。メニューの編集システムが完備されているため独自のメニューも容易に作ることができる。図形処理ソフトやユーザー・コマンド/ユーザー・プログラムとのインタフェース方式も統一されている。特にICONメニューは3次元ボタン(指示すると，あたかもボタンを押したように見える)を含む，使いやすくわかりやすいメニューになっている。

●データ構造の公開

より高度なカスタム化のために，図形データの内部構造をユーザーに公開している。

●データベース・マネージメント・システムとのインタフェース

VAXベースのシステムではネットワーク型のDBであるDMRSとのインタフェースが提供される。EWSではリレーショナルDBのInformix，Oracle，Ingresとのインタフェースが提供される。これにより，大量の属性データを扱うアプリケーションの開発が可能となる。

●ウィンド・システム

Environ Vというプログラミング機能を備えたUNIX上のウィンド・システムがあり，図形ソフトのウィンドに独自のウィンドや独自のメニュー・システムを組み込むことができる。

カスタム化の具体的な実施例

●A社(建築関係)

CADによる製図の効率をより上げるために，独自の建築製図機能をユーザー・コマンド言語とプログラム・インタフェース

により作成している。また，企画日影のアプリケーション・ソフトウエアも上記のツールを用いて開発し，インターグラフ・システムと連動させている。メニュー・システムを活用している点も特徴。

● B社(地図関係)

都市のすべての公共設備(電気，ガス，水道)データを図形と属性に分け，地図データベースに登録するシステムはカスタム化ツールを駆使して構築。特にデータベース・マネージメント・システムとのインタフェース部分に，データの保全と大量データ(数ギガ・バイト)の扱いを考慮したカスタム化が行われている。データ構造の公開が開発時に大きな助けとなっている。

● C社(機械関係)

カメラの電子制御系のフレキシブル基板を設計するアプリケーション・ソフトウエアをユーザー・コマンド言語とプログラム・インタフェースを使って構築。

● D社(エンジニアリング関係)

空調設備の実施設計を3次元で行い，施工情報までを取り出すアプリケーション・ソフトウエアをユーザー・コマンド言語とプログラム・インタフェースを使って構築。

カスタム化に対する方針

ユーザーとCAD/CAMメーカーの相互の利益とCAD/CAMシステムの健全な発展を考えると，カスタム化は現在の「独自のシステムを作り上げる」というレベルから「微調整して使いやすいようにする」というレベルに変化する方がよい。独自のソフトウエアを開発する時代から既存のソフトウエアをいかに利用するかの時代になりつつあるといわれるようになってきたのは，CAD/CAMシステムが真の普及期に入りつつある証であろう。この流れの中では，ユーザーは自らカスタム化を進めるよりもメーカーに対して既存のパッケージ・ソフトウエアを改良する要求を出した方がよい。また，メーカーもユーザーの要求に対してどこまでこたえていけるかが生き残りのポイントとなる。

当社は，これまではユーザーのカスタム化の要求に個別に対応してきたが，今後は要求をまとめてパッケージ・ソフトウエアに反映させる方式にも重点をおくことを考えている。ユーザーが大規模なシステムづくりを要求してきた場合は，コンサルテーションやサンプル・プログラムの提供などを中心とするサポートをこれからも実施していく。

問合せ先：武藤工業㈱　インターグラフ事業部営業部
〒153　東京都目黒区中目黒4-6-1
☎03(760)6111
開発元：Intergraph Corp.(米国)

CATIAのカスタム化機能

日本アイ・ビー・エム

CAD/CAMシステムが提供しているカスタム化の機能

CATIAが提供しているカスタム化のツールは，大きく分けると次の2つからなる（**図1**）。

● API（アプリケーション・プログラム・インタフェース）

● CATIAユーザー・インタフェース

以下に，おのおのについて説明する。

APIは4つのサブルーチン群で構成されている。

① CATGEOサブルーチン：CATIAデータ(モデル，アトリビュート，…)の読取り，書込み，修正，削除やCATIA資源の管理を行う。

② CATMSPサブルーチン：CATIA内部で使用している数学関連のルーチンである。このルーチンを使用することによって，既存のデータを使用してその交点を取ったり，ブーリアン・オペレーションを行ったりすることが可能である。

③ DIM（データ・インプット・マネージャ）：ユーザーがCATIAセクション中に，その対話手順の中で何を行ったかを取り出して解析する。

④ CGI（CATIAグラフィックス・インタフェース）：これを使用し，CATIA画面内にポップアップ・ウィンドを作成することができる。

APIは以上のバッチ・サブルーチン群をもって，ユーザー作成のバッチ・プログラムや後述のCATIAユーザー・インタフェースに組み込むことによって，対話式アプリケーション・プログラムを容易に作成することができる。

CATIAユーザー・インタフェースは2つのものからなる。

まず，IUA（Interactive User Access）コマンド言語がある。IUAはAPIのCATGEOやCATMSPを利用して，ユーザー開発のプログラムを対話方式で実現するインタプリタ言語である。対話機能はIUAファンクション内で実行され，ユーザー・プログラム登録リストの中から使用したいものを選択することによって起動する。またマクロ機能をもち，すでに標準で300以上のマクロが用意されているうえ，ユーザーが新規に登録することも可能である。このマクロはCATIAのどのセクション中からも呼び出せ，マクロ機能終了後は引き続き前のセッションに戻るようになっている。

もう一つは，GII（Graphic Interactive Interface）機能である。GIIはCATIAモジュールの開発用ツールで，API各サブルーチンをフルに利用することによって，ユーザー開発のアプリケーション・プログラムをCATIAのファンクションとして作成することができる。これによりユーザーは，CATIA環境内でCATIAと同じ対話方式を使用して独自の処理を行うことができる。

カスタム化の具体的な実施例

CATIAのカスタム化の例として，建設業の設計業務に密着した先進のユーザー・アプリケーション事例，鹿島建設が開発したREALS，SPACIOという2つの例を紹介する。

REALS は意匠検討，照明検討を行う際に非常に視覚的なイメージ（REAL なイメージ）を作成することによって支援するツールである。

CATIA の 3 次元モデリング機能で計画建築と背景，あるいは内観と照明などのモデルを作成する。このモデルに対してユーザー指定の材質と光源の情報を付加し，シェーディング・イメージを計算する。その結果は CATIA のピクチャー・データとして返すことができ，またこれらのイメージを連続して作成し，施主プレゼンテーションに活用できるアニメーション・フィルムを作成することもできる（**図 2**）。

SPACIO は，建築企画・設計の法規チェックを効率良く行い，計画地に対して最大建築可能空間を計算する最適設計のためのシミュレーション・ツールである。CATIA の優れた 2 次元機能（特にソリッドとブーリアン演算）を活用し，ユーザー・インタフェースに IUA を使用して CATIA の対話環境に完全に統合されている。そのため設計者は思考を中断することなく，一連の使いやすいオペレーションで企画設計を効率良く進めることが可能である（**図 3**）。

その他，重工業でのロボットに関するオフライン・ティーチングの例など，多数の実例がある。

カスタム化に対する方針

CATIA Ver. 3 の最大の特徴はオープン・アーキテクチャに

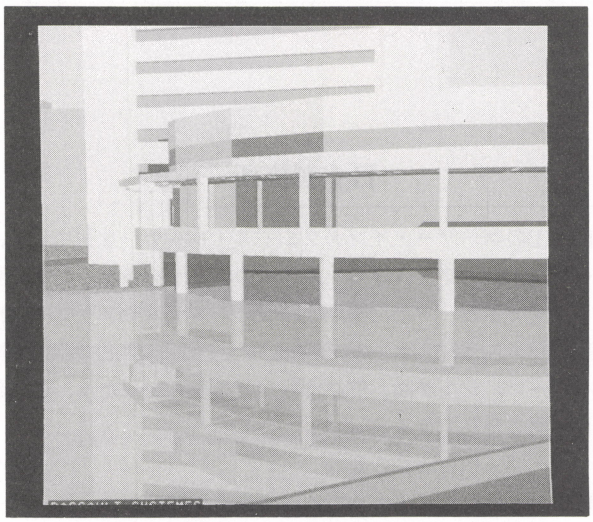


図　1

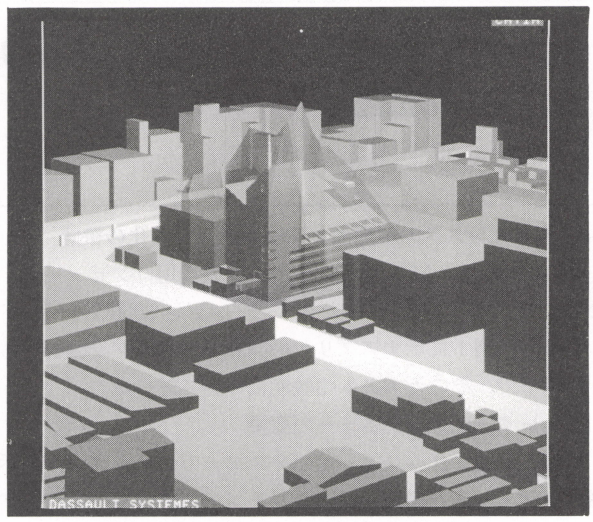
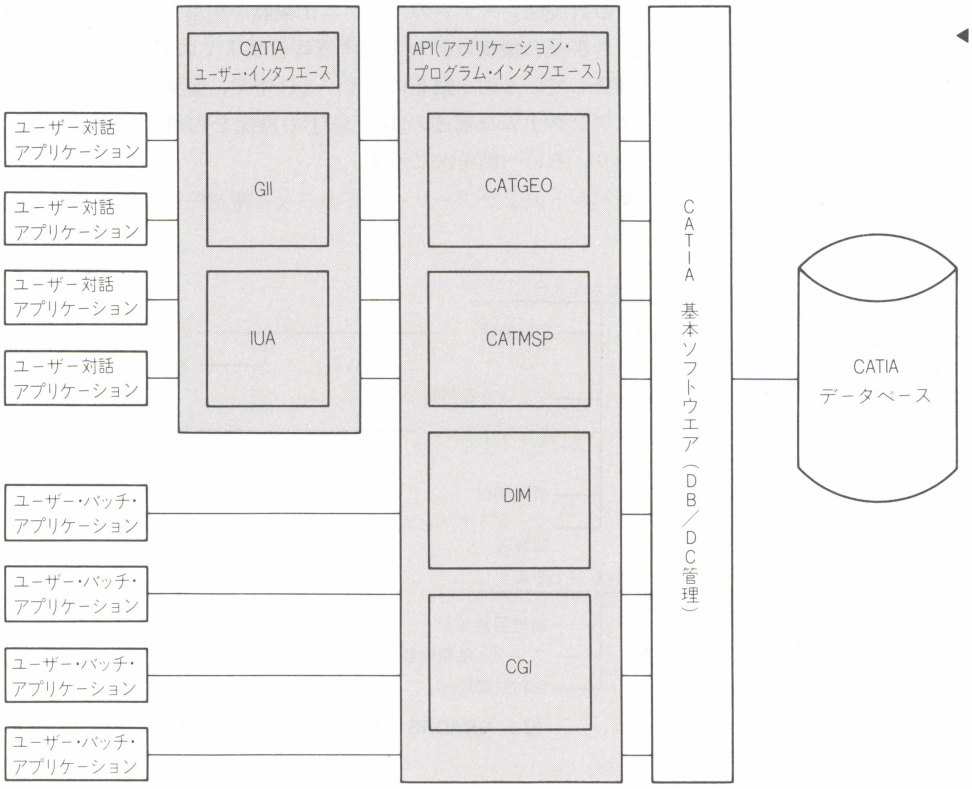


図 2　REALS の例



◀図 3　SPACIO の例

ある。オープン・アーキテクチャの採用により，ユーザーは次の点でカスタム化後も利益を受けることができる。

●単一のオペレータ・インタフェース（対話手順）の確保
●データ一元管理の確保：アクセス方式の統一，バッチ処理，対話方式で共通のデータを使用
● API の永続的サポート
●ユーザー作成プログラムの独立：CATIA から，OS（オペレーティング・システム）から

これらにより，ユーザーが CATIA 上に独自のアプリケーションを構築した後，たとえ CATIA がバージョンアップやリリースアップを行った場合でも，またハードウエア（機械）が機能向上した場合でも，それまでのユーザー・プログラムはなんら変えることなく使用することが可能である。

また，システムの基本的な機能（データベースの管理，マン・マシン・インタフェースの管理など）は CATIA の機能を利用することができるので，プログラムの開発が容易となる。

問合せ先：日本アイ・ビー・エム㈱　エンジニアリング・システム第二 E/S 推進センター
〒163　東京都新宿区西新宿 2-6-1
☎ 03(344)1311
開発元：Dassault Systems 社(仏)

日立グラフィックスシステム GRADAS のカスタム化機能

日立製作所

CAD/CAM システムが提供しているカスタム化の機能

GRADAS は，M シリーズ・ホスト型汎用コンピュータ，またはエンジニアリング・ワークステーション 2050G をハードウエアとし，**図 1** に示すアプリケーション・プログラム群からなる。以下で，そのカスタム化機能について述べる。

(1)　ユーザーによるメニューの変更

画面に表示されるコマンドのメニューのツリー構造，表示内容，パラメータの種類と個数などをユーザーが変更・追加することができる。また，シート・メニューの定義も自由に変更可能である。

(2)　マクロコマンド機能

マクロコマンドは，一連のコマンド群を組み合わせて 1 つのコマンドとして実行できるようにしたものである。マクロコマンドは対話で作図しながら，変数も組み込んで定義・編集することができる。本機能は設計製図システム，3 次元設計システム，ワークステーション型製図システムで用意されている。

(3)　ユーザー・ノウハウの組込み

① ユーザー作成コマンドの組込み

ユーザーが独自の専用コマンド処理ルーチンを作れるように，コマンド・パラメータの入力，形状データまたは属性データの参照・更新・作成，幾何計算などのユーザー・インタフェース・ルーチンを用意している。さらに，各コマンド処理ルーチンへ制御を渡すためのユーザー・コマンド出口ルーチンを組み込めるようにしている。この機能は，設計製図システム，3 次元設計システム，ワークステーション型製図システムなどで用意されている。

② ユーザー情報のライブラリ化

2 次元 NC 加工システムでは，加工手順，工具の選択，切削条件に関するユーザーのノウハウを，また，工作機械の仕様や NC テープ・フォーマットを指定したデータをライブラリに登録しておける。

(4)　ユーザー・プログラムによるデータの参照・更新

前項①のユーザー・インタフェース・ルーチンを利用して，ユーザー・プログラムからシステムのデータを参照したり，更新したりできる。また，論理回路設計システムやプリント基板実装設計システムでは，ファイルを介して部品の割付け状態や配線パターン，シルク図，部品情報などを取り出したり，逆に設定して返したりできる。

(5)　ユーザー独自部品の登録

プレス金型設計システムでは，標準金型部品の他，ユーザーが部品形状や加工情報，部品の取付け角などを定義でき，自由な金型構造を設計できる。

カスタム化の具体的な実施例

設計製図システムのカスタム化機能を利用して，メカトロニクス製品の機構設計から生産情報管理までの CIM システムを構築している例を**図 2** に示す。この中で，製図支援・解析支援サブシステムは前述の(1)〜(3)-①の機能を利用して実現されており，その一例を次に示す。

●ベルトおよびプーリーなどからなる搬送路自動作画機能：紙

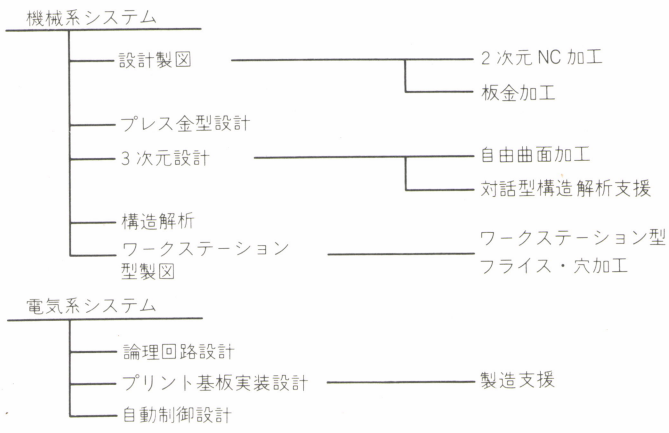


図 1　GRADAS 主要アプリケーション・プログラム

幣，伝票などの搬送路を自動作画するとともに，ベルトの接触角や長さを自動計算する。変更も容易にできる（**図3**）。

●リンク機構解析図作成機能：図形どうしが相互接触するまでの移動または回転挙動を解析し，作図する。また，図面管理サブシステムでは，前述の(1)，(3)，(4)の機能を活用し，図面の審査・承認署名のCAD化，データ・セキュリティの確保，図面の入庫や生産手配への情報出力などを実現している。

カスタム化に対する方針

●ニーズのレベルに応じたカスタム化機能

簡単に作れるマクロコマンドや標準値・形状の変更機能のレベルから，ユーザー・インタフェース・ルーチンを利用してユーザー専用のシステムを作るレベルまで，ニーズの深さに応じたカスタム化機能をそろえている。

●ワークステーション・ホストのネットワークによるCIMの構築を支援

エンジニアリング・ワークステーションとホスト・コンピュータでネットワークを構成し，解析などの負荷の重い処理やデータベース的役割をホスト・コンピュータにもたせたうえで，CAD/CAMシステムの情報を参照したり，逆にCAD/CAMシステムへ情報を与えたりする手段を提供することによって，ユーザーがホスト・コンピュータの生産管理システムや情報管理システムなどと連結して，トータルシステムCIMを構築できるようにしている。

問合せ先：㈱日立製作所　情報システム営業部
〒140　東京都品川区南大井6-27-18
☎03(763)2411

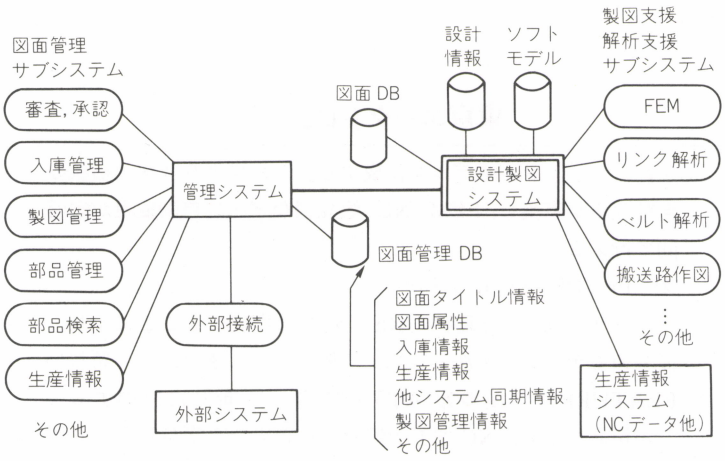


図2　メカトロニクス製品の設計・生産CIM化構成例

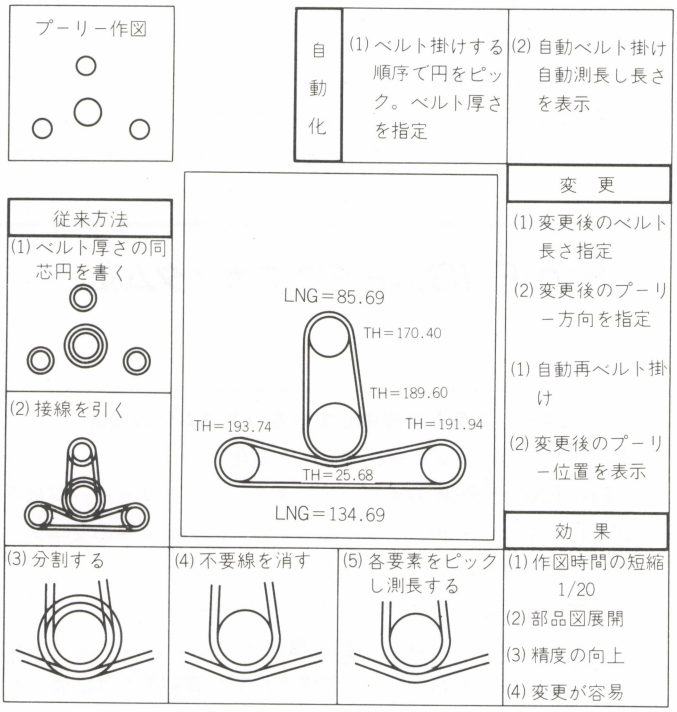


図3　搬送路自動作画機能と従来の標準機能の比較

CAD SUPERシリーズのカスタム化機能

機械系

アンドール

CAD/CAMシステムが提供しているカスタム化の機能

(1)　自由定義ができるメニュー(画面レイアウト)

設計者独自の操作環境を提供できるメニュー・マネージャをもち，メニューの形状，項目の変更が自由自在にできる。そして，デザインワークの目的により，プルダウン・メニュー，画面メニュー，タブレット・メニューなどを任意に組み合わせて使用することもできる。これらの変更はプログラムを毎回修正するのではなく，外部定義ファイルのみの修正で簡単に行える。

(2)　専用CAD言語の提供

カスタム化を容易にするために専用CAD言語「スーパーランゲージ」が用意されている。

このスーパーランゲージは，一般によく利用されているBASIC言語によく似た内容となっており，BASICを知っていれば簡単に使いこなすことができるようになっている。このスーパーランゲージではCAD SUPERの基本機能はすべて使用することが可能。また，ここで作成された追加機能は前述のメニューの任意の位置に登録でき，図面作成中に呼び出すことができるだけでなく，CADシステムを立ち上げることなく単体でも利用できるので，アプリケーションなどのオリジナル・ユーティリティを作ることができる。

カスタム化の具体的な実施例

(1)　画面メニューとタブレット・メニュー

当社では，CAD SUPER EX/Mの画面メニューとタブレット・メニューとを標準で提供している(ユーザーによる最初の設

定で切り換えて使うことが可能)。

しかしユーザーによって不要なコマンドがあったり，またレイアウトも希望通りにはなっていない場合もあるため，ユーザー側で変更して使用している例もある。また，コマンド名として使用している言語，メッセージなども社内で使われているものに合わせて変更している例もある。

(2) スーパーランゲージの活用

複雑な部品など通常の操作では数十オペレーションを要するものをスーパーランゲージで記述し，設計者と対話していくことにより少ないオペレーションでの作図を可能にしている。また，当社が標準では提供していない特殊な作図コマンドを組み込んでの使用例もある。

(3) 専用部品表の作成

シンボルには複雑な属性をもたせることができる。また，部品表の様式もユーザーの必要に応じて外部ファイルで定義することができるので，ユーザーの要望にフィットした部品表の作成が可能である。

カスタム化に対する方針

現状のカスタム化は，ユーザー自身が行っているか，あるいはソフト・メーカーが別注ソフトとしての開発を行い，この場合は高額の開発費を必要とするというのが一般的であった。これに対し，当社が開発したCAD SUPER EX CORE は，どのようなアプリケーションにも耐え得るCADソフトのコアとなる機能をもっている。そのため，今後はコマンド・レベルのカスタム化から全く業種の異なるアプリケーションの開発までこのCAD SUPER EX CORE をベースに行っていく予定である(コア部は変更することなく，外部ファイルの定義変更あるいはアプリケーションを作り替えるだけで異なったCADシステムとなる)。

問合せ先：アンドール㈱　システム営業部
〒103　東京都中央区日本橋富沢町5-5
☎03(639)1451

Pro/ENGINEERのカスタム化

東京エレクトロン

CAD/CAMシステムが提供しているカスタム化の機能

(1) メニュー，メッセージのカスタム化

Pro/ENGINEERのオンライン・メニューとメッセージはそれぞれテキスト・ファイルになっており，メニューの文字メッセージの中身を自由に変更することができる。

(2) 寸法値とCプログラム・インタフェース

寸法値のリレーションの拡張機能としてCプログラムのインタフェースが可能で，Cで作成される計算結果を寸法値に反映させ，形状をその値に従って変更することができる。

(3) Pro/FEATURE

Pro/FEATURE はオプション・パッケージで，FEATUREのマクロ化機能である。Pro/ENGINEERでサポートしているFEATURE(例：穴，スロット，フィレットなど)のマクロ化が可能で，かつそのFEATUREに付属する寸法値を実行時に自由に変更し形状を作成することが可能である。

(4) Pro/DEVELOP

Pro/DEVELOP はオプション・パッケージで，Pro/ENGINEERのカスタム化のためのサブルーチンを用意しており，

①メニュー，メッセージ，ユーザー入力の機能
②図形データベースのダイレクト・アクセス機能
③オブジェクト・ベースのデータベース登録機能
④Pro/ENGINEER上で稼働する機能(Pro/EXEC)

などをもっている。

カスタム化の具体的な実施例

(1) プログラムによる形状の自動変更

Pro/ENGINEERで作成されたソリッドの自由曲面部分と別のサーフィス計算システムで計算された新しい点群データを読み取り，その点群データに基づいて自動的に曲面部分が変更されたソリッド・モデルを作成する。これにより3次元形状の自動変更が実現し，FEM, NC, 組立図などへの自動変更が短時間で行える。

(2) Pro/DEVELOPの利用

ユーザー独自のFEATUREを新たに作成することにより，従来のモデリング手法が大幅に短縮される。また，この新しいFEATUREに付属情報(ATTRIBUTE)を新たに付加することが可能なため，NC加工情報として必要な文字情報をソリッド形状と結び付けてデータベースに登録し，その出力データをもとに自動NC加工処理をすることができる。

カスタム化に対する方針

(1) 機能改善要求

Pro/ENGINEERの現機能に対する改善要求については，ユーザーのニーズをもとに積極的に実施している。

(2) 日本化機能

米国製品に対してよくいわれる日本化については，PTC社と共同で作業を進めており，メッセージの日本語化，日本語注記入力機能，JIS寸法線表示，日本語マニュアルなどをリリースする予定である。

(3) プロジェクト

大幅な機能改善や新しい技術の組込みなどのプロジェクト・ベースの開発については，そのつどプロジェクト・ベースとして取り扱っており，見積りし，条件などをつめて実施している。

問合せ先：東京エレクトロン㈱　CIMシステム部
〒163　東京都新宿区西新宿1-26-2
☎03(344)8904

開発元：Parametric Technology Corp.(米国)

オープンシステムとしてのEUCLID-ISの開放度

マトラ・データビジョン

CAD/CAMシステムが提供しているカスタム化の機能

(1) EUCLID-ISのカスタム化機能

- ESL言語：EUCLID-ISのアプリケーション・プログラムを開発するための言語で，FORTRANプログラムから使用できるサブルーチンのライブラリからなっている。図形や文字，寸法の作成，編集，デコード(作成の反対に，情報を取り出す操作)，図形演算，グラフィックス機能，データベース・マネージメント機能をはじめ，EUCLID-ISのもっているほとんどの機能をプログラムから呼び出して使うことができる。
- インターラクティブ・モニター：EUCLID-ISの対話処理をする部分で，グラマーとよばれる文法を用いて画面上のメニューや高度な対話処理を設計できる。また，オペレータ・インタフェースを統一するので，誰がプログラムを作成しても自然にEUCLID-ISの操作環境に適合する。例えば，入力エラーの扱いや，プロンプト(入力要求メッセージ)，エラーメッセージの表示，データの検索手順，ファンクション・キーの使い方，コマンドのリカバリー方法などが統一され，操作者に違和感のない操作方法となる。
- FORTRANインタプリタ：ESL言語のインタプリタ版で，メニューやコマンドを使っての作業中にプログラムを作成して利用できる。
- マクロコマンドモード：コマンドやメニュー操作をプログラム化できる。
- パラメトリック・デザイン：寸法を入れることによって形状を変化させることができる標準部品を作成することが可能。
- アプリケーション・パッケージ：基本モジュールだけでなく，アプリケーション・パッケージに対してもESL言語がサポートする。
- その他：システムのデフォルト値の設定，ダイナミック・スクリーン・メニュー，タブレット・メニューのレイアウトなどがある。

カスタム化の具体的な実施例

小規模な例，例えばメニューのプロンプトの変更，デフォルト値の変更，標準部品の作成，ユーザー・アプリケーションの開発などは多くの企業で当たり前に行っているので新鮮味がないため，特にEUCLID-ISの内部のデータ構造と深くかかわった例を紹介する。

あるヨーロッパの研究機関では，6万点に上る部品のソリッド・モデルを作成している。これらの部品を使用した装置は最大5階層のアセンブリとなる。ある装置は約800個のハーフコンポーネントからなり，各ハーフコンポーネントは平均40個のシステムにより構成される。ここでは，少なくともハーフコンポーネントを組み合わせた図面を作成する必要がある。また装置は最大30kmにも及ぶので，データはメートル単位とミリメートル単位のものがある。これをEUCLID-ISのラージアセンブリ機能で表示するのは困難であった。そのため次のようなプログラムを開発した。

巨大な装置の図面を作成する。その際に，

- 各部品の長さの単位を自動的に合わせる
- 表示すべき部品だけを記述したリストを参照する
- 下位レベルの部品をどの層まで表示するかを指定する
- 部品の配置を同時に行える
- 他の表示データをインポーズできる
- 結果をEUCLIDのLASS機能と同じデータ構造(ディスプレイリスト)に作成する

というプログラムである。

カスタム化に対する方針

システムの開放とユーザー資産の保護という課題は，ある意味で矛盾する命題を背負っている。すなわち，システムの開放をシステムの核の内部まで進めてしまうと，メーカーによるシステムの能力の改善(バージョンアップ)が行われたときにユーザー・プログラムが追従しきれなくなるという問題があるからである。

ユーザーのプログラム資産の保護を図りながらシステムをできるかぎり開放していくためには，ユーザー・プログラム・インタフェースというチャネルを介して，内部の情報を可能なかぎり普遍的なかたちにモディファイして提供していくという，地道な努力を積み上げていく以外にはないように思う。

また，データ構造に関しては完全にオープンであり，ユーザーが十分な知識をもつならその情報をすべて利用できる。

将来的には，部分的にユーザーがデータ構造を決めることもできるようにする予定である。

また，マニュアルなどの一層の充実により，実質的な開放度を高めることも重要と考えている。

長期にわたって安定的に供給できる部分については，可能なかぎりオープンにしてカスタム化をサポートしていきたい。また，それが可能となるアーキテクチャを追求している。そして，それを実質的に可能にするサポート力の強化に努力していきたいと考えている。

問合せ先：マトラ・データビジョン㈱　営業部
〒170　東京都豊島区東池袋3-1-1
☎03(988)1035
開発元：Matra Datavision(仏)

HP デザインセンターMEシリーズのカスタム化機能

横河・ヒューレット・パッカード

CAD/CAMシステムが提供しているカスタム化の機能

HP デザインセンター ME シリーズのカスタム化機能は,「カスタム化の容易さ」という言葉に集約される。各 CAD/CAM ベンダーは「カスタム化できる CAD/CAM システム」をうたっているわけだが，そのカスタム化機能はともすれば導入側にとって莫大な経費と時間のかかるシステムになってしまう。

ME シリーズは使いやすいシステムとして広くユーザーに受け入れられており，カスタム化機能もその例外ではない。当社のユーザー・アンケートでも,ME シリーズ導入のポイントにシステムのカスタム化の容易さが必ずあげられ，導入後も 90%以上のユーザーがなんらかのカスタム化を行っている。

ME シリーズのソフトウエア構成を簡単に示すと**図 1** のようになる。HP-UX(UNIX)オペレーティング・システム上で稼働し，コアシステムは基本モジュール，データ・マネージメント(管理)モジュール，バリエーション・デザイン・モジュールで構成される。

カスタム化の種類，ツールとしては，

- スクリーン・メニュー，タブレット・メニュー，キーボードのカスタム化をサポートするシステムマクロ
- ユーザー作成コマンドの付加，作図手順の一体化，自動設計をサポートするユーザーマクロ
- NC 自動プログラミング，FEM 解析ソフトウエア，他の CAD システムへのデータ転送において独自の機能をサポートする MI(モデル・インタフェース)
- データベースへのアクセス項目を変更・定義するデータ・マネジメント・システムのカスタム化
- シンボル，テキスト書体のカスタム化

などがある。ユーザーが他のプログラミング言語(C, Pascal)で作成した計算プログラムを ME シリーズ内に取り込み，カスタム化することも可能である。また，日本語入力機能におけるユーザー辞書登録機能も一つのカスタム化機能といえる。

カスタム化の具体的な実施例

(1) メニュー・レイアウトのカスタム化

ME シリーズは,スクリーン・メニューを画面上の任意の位置に定義できる。"MENU_LAYOUT"コマンドを使い，ME シリーズのエディタ内でキーボードを利用して定義を行う。このように定義すると,ME シリーズの右下画面にメニュー・レイアウトが作成される。ユーザーは同様の手順で右上，左下などにメニュー・レイアウトを定義でき，メニューの大きさなどを自由に作成できる。

(2) メニュー・コマンドのカスタム化

ME シリーズは標準コマンドやユーザー作成コマンドをメニュー・レイアウト上に自由に定義できる。

MENU BLACK WHITE '点' 'POINT' 3 1

とキーボード上で入力すると，メニュー・レイアウトの上から3番目，左から1番目の部分に"点"という表示でバックが白，文字が黒で出る。コマンドは"POINT"という標準コマンドが定義される。POINT の代わりにユーザー作成コマンドを定義することも可能である。同様の操作で定義したのが**図 2** である。また，**図 3** の下段のメニューはユーザーがこの機能を利用し，作成したカスタム化例である。

(3) ユーザー・コマンドの作成

簡単なプログラミングでユーザーが独自のコマンドを作成できる。このコマンドは直線を等間隔に任意の数に分割するコマンドで，ユーザーは"A_line_dividing"というプログラム名を使用し，前述のように

MENU BLACK WHITE '線の分割' 'A_line_dividing' 2 1

とすれば,メニュー・レイアウトの上から2番目,左から1番目の部分に"線の分割"と表示され,このコマンドを選択すると"A_line_dividing"のプログラムが実行される。

カスタム化に対する方針

CAD/CAM システムの導入を考える側にとって最も重要になるポイントは，そのシステムが自分たちの要求する仕様にいかに近いシステムであるか，という点であるが，ユーザーの要求する仕様に対し 100%対応できる CAD/CAM システムは残念ながら存在しないだろう。

このような状況の中，ベンダー側の対応としては，カスタム化できるシステムやツールを提供するということだろう。しかしここで見落としてはならないのは，ユーザーがカスタム化を考えるとき，ツールがそろっていても使えなければ何の意味ももたない点である。YHP ではユーザーのカスタム化のレベルに応じ，それぞれのサービスを提供している。

(1) トレーニング・サービス

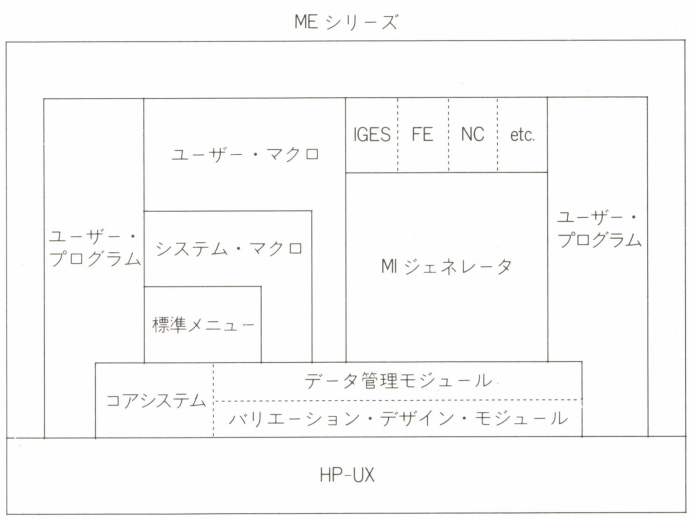


図 1　ME シリーズのソフトウエア構成

MEシリーズのトレーニング・コースの中にカスタム化のコースが設けられている。ここではメニュー，コマンド，シンボル，書体などの定義方法が解説される。多くのユーザーは，このコースを受講することにより基本的なカスタム化が行える。

(2) コンサルティング・サービス

カスタム化をはじめとする各種コンサルティングに対応するため，SE部門が各支社に設けられている。機械系CAD専任のSEにより，ネットワーク環境，データベース環境を考えたカスタム化，また，バージョンアップ時におけるカスタム化への対応などを中心にコンサルティングを行っている。

(3) プロジェクト・サービス

MEシリーズの大幅な仕様変更，それに伴う特注ソフトウエアの開発は，プロジェクト・センターを中心に対応している。YHPで提供する電子系CADシステムにとどまらず，他社CADシステムをも含めたプロジェクトの中でのカスタム化，そして，その導入手順などがプロジェクト・センターによってサポートされる。

YHPでは，現在，MEシリーズのOEM開発を行っている。今後この分野に対しさらに積極的に力を入れるため，OEMユーザーに対してのカスタム化ツール，サポートの強化を進めていく方針である。

問合せ先：横河・ヒューレット・パッカード㈱　第1コンピュータ部門CAME課

〒168　東京都杉並区高井戸東3-27-21

☎03(335)8234

開発元：Hewlett-Packard Co.(米国)

```
MENU BLACK YELLOW '       作　図 1' '' 1 1
MENU BLACK WHITE '  点' 'POINT' 3 1
MENU BLACK WHITE '上書き' 'I_overdraw' 3 2
MENU BLACK WHITE '  線' 'LINE TWO_PTS' 4 1
MENU '２点の線' 'LINE TWO_PTS' 4 2
MENU '連続線' 'LINE POLYGON' 5 1
MENU '水平線' 'LINE HORIZONTAL' 5 2
MENU '鉛直線' 'LINE VERTICAL' 6 1
MENU '平行線' 'LINE PARALLEL' 6 2
MENU '垂直線' 'LINE PERPENDICULAR' 7 1
MENU '円の接線' 'LINE TAN_PT' 7 2
MENU '２円の接線' 'LINE TAN2' 8 1
MENU '角度と長さ' 'LINE PT_ANG_DIST' 8 2
MENU '長方形' 'LINE RECTANGLE' 9 1
MENU BLACK WHITE '  円' 'CIRCLE CENTER' 10 1
MENU '中心と半径' 'CIRCLE CENTER' 10 2
MENU '３　点' 'CIRCLE THREE_PTS' 11 1
MENU '半径２要素' 'CIRCLE TAN2_PT' 11 2
MENU '同心円' 'CIRCLE PARALLEL' 12 1
MENU '直　径' 'CIRCLE DIAMETER' 12 2
MENU BLACK WHITE '円　弧' 'ARC THREE_PTS' 13 1
MENU '３点の円弧' 'ARC THREE_PTS' 13 2
MENU '半径と角度' 'ARC CEN_RAD_ANG' 14 1
MENU '中心と２点' 'ARC CEN_BEG_END' 14 2
```

作　図 1

点	上書き
線	２点の線
連続線	水平線
鉛直線	平行線
垂直線	円の接線
２円の接線	角度と長さ
長方形	
円	中心と半径
３　点	半径２要素
同心円	直　径
円　弧	３点の円弧
半径と角度	中心と２点

図2　コマンド定義例

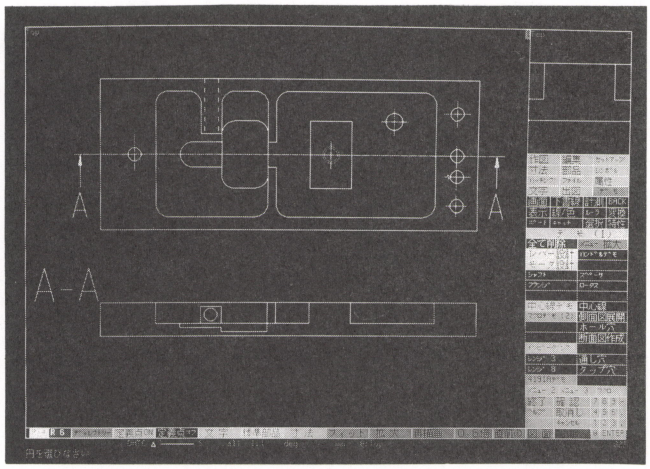


図3　ユーザー作成メニュー

セジュール-Aの推論設計によるカスタム化

電子・電気

シャープ精機
シャープシステムプロダクト

CAD/CAMシステムが提供しているカスタム化の機能

ユーザーがカスタム化を行う条件としては，頻繁に類似(相似形)のオペレーションを行う必要が生じることである。したがって，ある程度一定のパターン化されたオペレーションがある場合に，そのオペレーションをCAD/CAMシステムの一つのコマンドにできれば便利である。セジュール-Aのカスタム化ツールとしては次の2つの機能がある。

●パーツ登録

CAD機能を使って作成した図形をパーツ(部品)として登録できる。登録したパーツは直接パーツ名を入力するか，メニューで指示することにより，指定した位置と倍率で呼び出すことができる。よく使うユーザー独自の部品，例えばボルトなどを登録しておくと便利である。

●推論設計

パーツ登録機能では，定型的な形状しか対応できない欠点がある。そこで，構造的には同一形状であるが寸法が相似形ではない場合や，一部分のみ異なった形状がある場合には推論設計機能が便利である。この機能は，ユーザーがCAD機能を使って作成した図形をそのまま変数化して登録できる。また，登録された変数化部分をグラフィック・ディスプレイで対話的に追加・修正する知識登録機能もある。知識登録はC言語によるif～then方式で記述していくが，文法チェックなどはシステムが行うためユーザーはなんら特殊言語の知識を必要としない。このようにして登録したパラメトリック設計ファイルをサブルーチン化して階層構造にし，複雑な機構部品の設計を行ったり，ファイルにより他言語(FORTRANなど)とのデータの受渡し

も可能となっている。推論設計の実行は知識登録したパラメトリック・ファイル名を入力すると，システムは変数化した寸法を尋ねてくる。これらを入力すると，グラフィック・ディスプレイに寸法線の入った図形が出力されてくる。

推論設計機能の最大の特徴は，CADで作った図形をそのままパラメトリック化し登録できることである。したがって，ユーザーはプログラミングを行う必要は全くなく，誰でも簡単にカスタム化が可能である。

図1に推論設計の手順を示す。

カスタム化の具体的な実施例

セジュール-Aは金型自動設計の機能があるので，金型メーカーに多く使ってもらっている。したがって，カスタム化も金型設計に対するものが多い。その一例としてモールドベース設計のカスタム化がある。これは，成形品の大まかな形状と金型タイプを対話形式で入力すると自動的にモールドベースが設計され，市販のモールドベースが寸法線の入った図面として出力されるというものである。**図2**に出力例を示す。

カスタム化の手法は，前述のようにCAD機能を使い適当な大きさのモールドベース図を描く。そして，このモールドベースの各寸法を変数化し，成形品の大きさとモールドベースの各変数値関係をif～then方式で定義していけばよい。これによって，成形品形状から自動的にモールドベース図面を出力するユーザー独自のコマンドを確立できる。

カスタム化に対する方針

当社は，CAD/CAMシステムは専用化すればするほど効果が上がると考えている。したがって，ユーザー・カスタム化の機能については最優先開発項目としている。また，どのような業種のユーザーであっても対応できる柔軟さや，特別な言語を必要としない簡便さもさらに追求していく考えである。また，現在の当社の推論設計の推論部分はif～thenによるものであるが，将来は本格的なAIを用い，知識登録の容易さや推論の並列処理の多様性の向上を図りたい。つまり，ユーザー独自のエキスパート・システムを誰でも簡単に構築できるよう目指したい。

当社のカスタム化機能は，CAD機能で作成した図形をもとにパラメトリック化するようになっているため，大幅な変更が生じた場合でもユーザーが簡単に対応できる。

問合せ先：シャープ精機㈱　営業部
〒581　大阪府八尾市跡部本町1-41-33
☎0729(91)0681

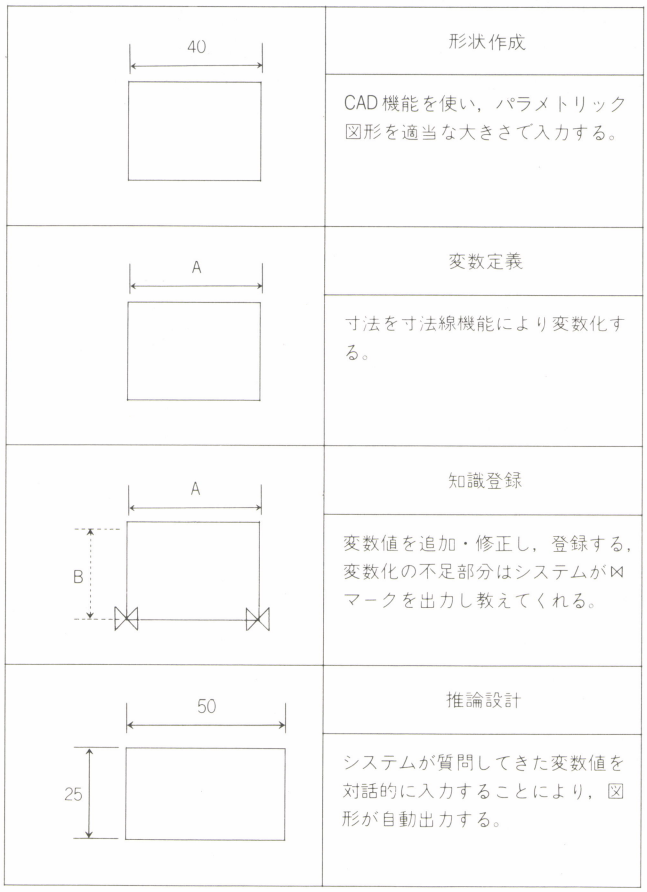


図1　推論設計の手順

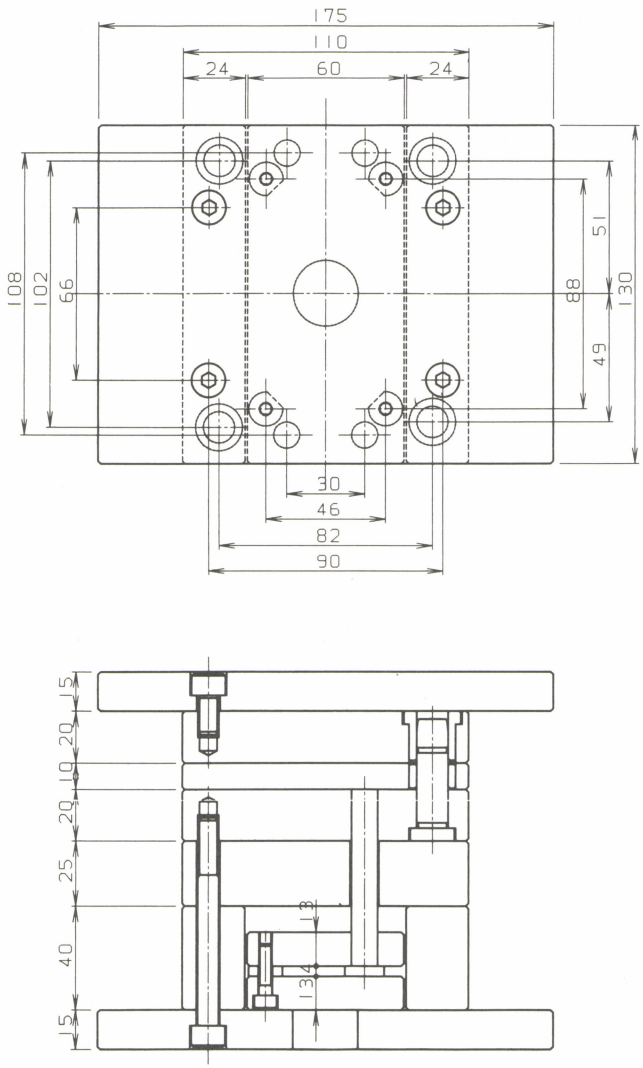


図2　モールドベース図面の出力例

プリント基板用CAE,CAD/CAMネットワークステーション CR-3000のカスタム化

図　研

CAD/CAMシステムが提供しているカスタム化の機能

図研のプリント基板用CAE，CAD/CAMネットワークステーションCR-3000は，ユーザー・ノウハウを反映した利用環境になっている。

(1)　CADのコマンドをユーザーが簡単につくれるZCL

ZCLとはZuken Command Languageの略で，CADのコマンド作成用の簡易言語である。

簡易言語とは，FORTRANやCなどのプログラム言語の経験のない者でも利用できるよう目的にそって簡易化した言語である。ここでは「CADのオリジナルコマンド作成」ということに対応している。簡単な例としては複数コマンドを一命令に定義することがある。これは使用するユーザー環境，運用により必ず一連の連続した命令をする場合に便利である。例えばパターン設計終了時に必ずバックアップ操作をしたい場合，「終了」という命令にこの一連の操作をもたせておくとオペレーションは楽になり，バックアップを忘れることもなくなる。

(2)　データベース・アクセスライブラリ

こちらは簡易言語と違い，FORTRANやCなどの言語を使いこなしているユーザー向けのツールである。

CR-3000のデータベースをユーザーがFORTRANやCなどの言語によるプログラム中に容易にアクセスできるよう，必要なライブラリをまとめたものである。

設計ファイルの読出しやファイルへの書込みを行うことになるので，プログラム言語とCR-3000のデータ形式の完全な理解が必要である。

カスタム化の具体的な実施例

CR-3000は64年1月に発売されたもので，現在のところ具体的な実施例は記載できるものがない。

カスタム化に対する方針

システムの拡張，最適化(カスタム化)がユーザー自身の手で容易にできることは，今後，CAD/CAMシステムの不可欠な要素になると思う。

ユーザーごとに設計基準や設計フロー，オペレーティング方法は異なるであろうし，システムを使いこなしていく過程においてノウハウも蓄積されていくので，ユーザーはより自分に合ったシステムづくりを行い，設計の効率向上や均質化を行うことになる。

CAD/CAMのメーカーとしてはユーザーニーズの取入れを積極的に行っていくが，「カスタム化」そのものがユーザーニーズとなる日は近いと思う。

問合せ先：㈱図研　システム営業部
〒222　神奈川県横浜市港北区新横浜3-1-1
☎045(473)6866

E-CADのカスタム化

ワコム

CAD/CAMシステムが提供しているカスタム化の機能

ワコムの電気系CAD「E-CAD」シリーズおよび機械系CAD「ϕCAD」は，ともにパーソナルコンピュータをベースとしながらも優れた設計環境を提供している。

その結果，多くのユーザーからワークステーションやミニコンピュータ以上のCADの入力マシンに採用したいとの要望があり，それに対してはデータベース・フォーマットを公開し，ユーザーサイドでのカスタム化を可能としている。また条件によってはソース・プログラムの公開も可能である。

さらに，ユーザー・プログラミング言語「MISTY」を提供，これを使用することによりユーザー独自のカスタム化CADの構築が可能である。

またワコムでは，カスタマイズセンターにSEを配置し，ユーザーと一体となったCADシステムの構築にも積極的である。

カスタム化の具体的な実施例

E-CADのユーザーであるA社では，ワコムの公開したデータ・フォーマットから独自にCADAMとのデータ変換プログラムを作成し，効果的な利用を行っている。すなわち，E-CADで見積り用の概略図面の入力を行い，JOBの正式にスタートしたものについてもE-CADデータをCADAMにリンクし，正式に図面として仕上げている。

また，このようにユーザーサイドで開発されたE-CADの周辺プログラムは，E-CADのアプリケーション・ソフトとして他のE-CADユーザーに提供することも可能である。

問合せ先：㈱ワコム　第2営業部
〒171　東京都豊島区池袋4-394-1
☎03(985)7911

建築・土木分野の企画から実施までを支援する本格派 CAD システム GDS のカスタム化

建築

エイ・アール・シー・ヤマギワ

CAD/CAM システムが提供しているカスタム化の機能

GDS(Graphics Design System)は，建築・土木・エンジニアリング関連分野のための本格的な汎用設計製図システムである。GDS は，オブジェクト指向の高性能データベース構造と製図のための強力な入力・編集機能，およびユーザー独自の運用を可能にするカスタマイズ機能を特徴としている。

(1) 操作性を向上させるカスタマイズ機能

- GDS ライブラリ機能：複数のユーザーが共用できる共通部品，標準部材などを収めてあるドローイング・ファイルである。ユーザー独自で自由に作成・登録をすることができる。
- メニュー・システム機能：豊富に備えられているコマンドやユーザーが作成するデータ，マクロファイル，入力ファイル，プログラムなどをメニューとして簡単に登録できる。
- ラインスタイル/キャラクタスタイル機能：より精緻な図面表現を行うために多数のスタイルを提供しているが，さらにユーザー独自に作成・登録も容易に実行できる。

(2) BASIC言語による自動設計，ユーザー・プログラミング(**図1**)

設計製図の中で演算や条件のチェックが必要な作図のため，BASIC 言語を用いて GDS 上でユーザー・プログラムを組むことができる。パラメトリック図形の作図や，演算・判断を必要とする設計作業のためにユーザー独自のコマンドを作成し，作業能率を向上させることが可能。また，GDS のデータベースと密接に連携し，データ検索・更新を行う自動設計システムを構築することができる。求積図作成・建具表作成・断面リスト作成などの専門アプリケーションを作成できる。

(3) GDS 属性のデータシステム(**図 2**)

GDS のデータベースは，図形情報と属性情報を密接に関連させて蓄積し，仕様に関する文字情報やコスト・数量などの数値情報を扱うことができる。数値情報を四則演算して加工でき，ユーザーは自由な書式で概算コストや部品一覧表などの積算情報を出力することができる。

(4) データの拡張，EPI(外部プログラム・インタフェース)

EPI は，ユーザー・プログラムと GDS がデータやコマンドを交換するためのリンクを提供するシステムである。FORTRAN，Pascal，C などの言語プログラムをサポートし，リンクを通して GDS のあらゆる機能を利用できる。アプリケーションの開発の際に多数提供されている GDS との対話用サブルーチンを用いて開発コストを抑えつつ，ユーザー独自のアプリケーションを構築することができる。

カスタム化の具体的な実施例

■施工図作成システム(**図 3，4**)

施工図作成システムは，GDS ユーザー，大日本土木(1986 年 3 月導入)が開発したプログラムである。GDS の BASIC システムにより，GDS の強力な図形入力・編集機能を利用して機械的な製図作業の部分を最大限に自動化し，施工図を迅速かつ容易に作成する。

- 一連の操作手順がメニュー化され，機能と入力手順がビジュアル化されているので，初心者にもすぐ操作できる。
- タブレットはわかりやすい図形と日本語による表示である。
- GDS で行うオブジェクトやラインスタイルはシステム内で設定しているので，オブジェクト名やラインスタイルを気にすることなく後の作業にも便利な整合性のある設定ができる。
- 頻繁に使用する定形の図形やシンボルは，すべてパラメトリックに描けるようにマクロ化され，作図時間の短縮が可能。
- 杭伏図，屋根伏図，基礎伏図，一般断面図，一般伏図，タイル割付図などがある。

カスタム化に対する方針

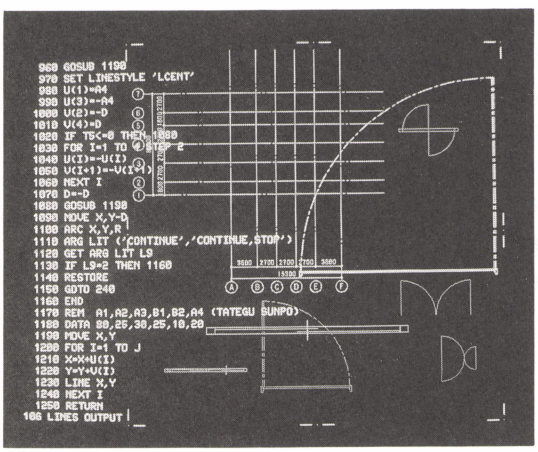


図 1　BASIC システム

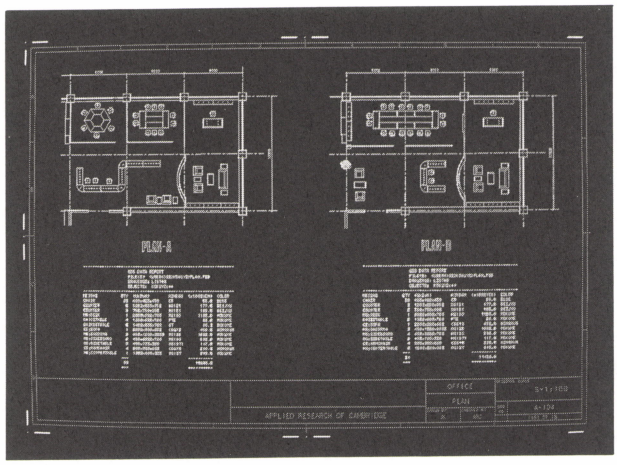


図 2　属性データシステム

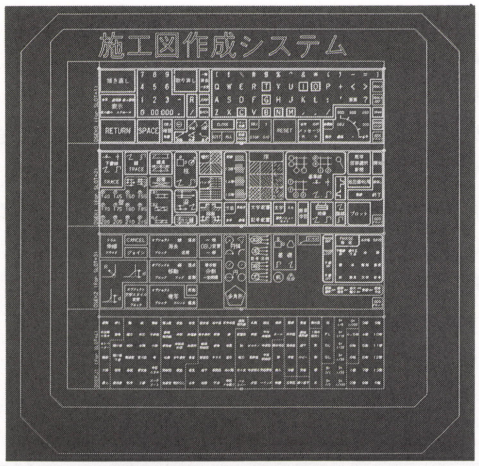


図3　施工図作成システムのメニュー

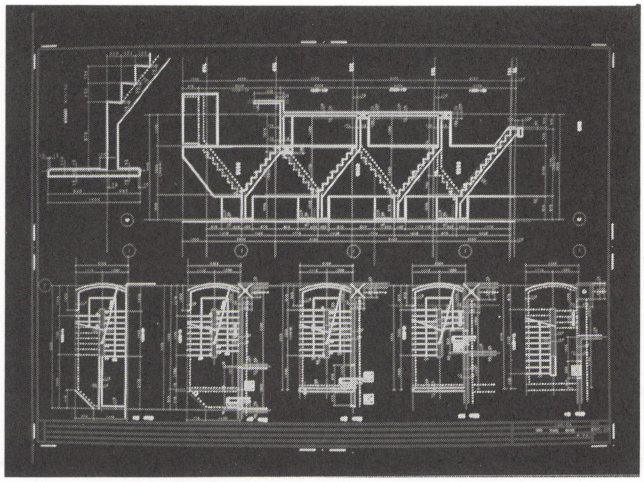


図4　施工図（提供：大日本土木）

GDSで用意しているユーザー・プログラム・インタフェースを用いれば，かなりの範囲のニーズにこたえることができる。また，外部のデータベース(例えば，リレーショナル・データベース)とリンクしてシステムを組んだり，他のプログラムとのデータ交換をすることで，さらに広範なシステム拡張が可能である。現在，世界で800以上のユーザーサイドで同一のシステムが稼働しているので，信頼性の高いシステムグレードアップ・メンテナンスをしていくためには，特別な別プログラムに改造すべきではないと考える。全体のシステム拡張を考え合わせながら，ユーザーの要望を取り入れて改良していく方針である。

問合せ先：エイ・アール・シー・ヤマギワ㈱　営業部
〒151　東京都渋谷区千駄ヶ谷1-7-16
☎03(405)7811
開発元：Apllied Research of Cambridge Ltd.(英国)

ダイナパース2/3を使ったカスタム化例

ダイナウェア

CAD/CAMシステムが提供しているカスタム化の機能

(1)　ユーザー側で独自に行えるカスタム化

- データファイルの内部構造の公開：ユーザーが，ダイナパースで作成したデータを別のソフトウエアで読み込み，加工できる。また逆に，別のソフトウエアでダイナパース用のデータを作成し，ダイナパース上で編集・加工ができる。
- DXF形式のデータを入出力できるオプションを別途販売：DXF対応のソフトウエア(例えばAutoCAD)との間でデータの互換が可能。
- 多種類の周辺機器への対応：ダイナパース3では，タブレット，プロッタ，プリンタ，フレーム・バッファなどの周辺機器への対応条件を，ある程度ユーザー側で独自に設定できる(ダイナパース2の場合は制限付き)。
  ①通常のタイプの周辺機器であれば，通信条件の変更はソフトウエア上においてワンタッチで可能。
  ②コマンドの異なる周辺機器にも，コマンドの内容を記述したテキスト・ファイルの変更により対応可能。
  ③特別なコマンドをもつ周辺機器の場合も，ユーザーが独自にドライバ・ソフトを開発し，対応させることが可能(ただし，別途契約が必要)。

(2)　ダイナウェア側で行っているカスタム化

ユーザーの目的・要望に合わせたソフトウエアの仕様変更，機能追加などを随時別途費用にて実施。

カスタム化の具体的な実施例

(1)　公開されているデータ・フォーマットを利用したユーザー事例

- 地図から等高線を入力し，ダイナパースの敷地のデータとして利用できるシステムの開発。
- 大型機で行った構造計算用の3次元データをダイナパース用のフォーマットに変換して表示する。
- ダイナパースのデータを大型機の日照計算用のデータに変換し，大型機上で日照計算を行う。
- ダイナパースで作成したデータを大型機上で作動する3次元とCADソフト上に読み込み，利用する。
- ダイナパースで作成したデータを，3次元ワークステーション上で作動する3次元レンダリング用ソフト上に読み込み，利用する。

(2)　DXF形式に変換したデータを利用したユーザー事例

- AutoCADで家具の平面配置プランを作成し，これをAutoCADの3次元用データとしてDXF形式で登録した後，ダイナパースで読み込んで家具の配置の3次元プレゼンテーションを行う。
- ダイナパースで作成した建物の3次元データのうち，任意の平・立面図をDXF形式で登録し，これをAutoCADで読み込

み，寸法線，ハッチングなどを入れて利用する。

(3) ダイナウェア側で行ったカスタム化事例

●ダイナパースに部品の積算見積機能を追加し，店頭で3次元および積算見積シミュレーションを同時に行えるシステムの開発。

●アニメーション撮影用のシステム開発。

①パーソナルコンピュータと16mmカメラを連動させ，アニメーションを作成するためのシステムの開発。

②3次元ワークステーション上で，コマ撮りVTRをコントロールし，アニメーションを作成するシステムを開発。

●システムキッチンのレイアウト用のシステムを，他のシステムハウスと共同で開発。

カスタム化に対する方針

●今後もデータ構造をユーザーに公開する。

●ユーザーが独自にソフトウエアを変更，機能追加できる部分を増やしていく。

●ソフトウエアの仕様の大幅な変更を必要とするようなケースに対しても，別途，別料金で対応していく。

問合せ先：㈱ダイナウェア　編集部

〒563　大阪府池田市石橋1-9-13

☎0727(62)8173

IRISシステムのカスタム化

CAE

エイ・アイ・テクノロジー

CAD/CAMシステムが提供しているカスタム化の機能

現在，CAD/CAM/CAEは急速に普及，パーソナル化が進み適用業種の広がりがみられる中，統合CAEシステムIRISは，構造解析のみならず流動・磁場など種々の解析業務をサポートできるよう，開発当初から「カスタム化」のためのツールの開発を手がけてきた。以下に主なツールをあげてみる。

●メニュー設定

IRISシステムのメニューはコマンド・ファイルに定義され，解析プログラムに合わせて修正が可能である。プリ・プロセッサでは拘束条件，物性条件など解析プログラム(構造，流動，磁場など)に合わせてメニューを設定できる。ポスト・プロセッサでも同様に，解析条件に合わせたタイトル・メニューの設定ができる。

●知識ベース

IRISシステムは知識ベースを用い，形状生成・データ変換作業の効率化が図れるように設計されている。「DKNOW」というモジュールがユーティリティとして用意されている。

if～thenルール，多岐選択，四則演算入力機能などをもつ独自の文法で定義された知識ベースにより，形状生成時には材料・荷重・拘束データやパラメトリック図形などの作成に，また，各解析プログラムのトランスレータではコントロール作成や要素変換などに利用されている。ユーザーは知識ベースを自由に変更・追加することができ，ユーザー独自のコンサルティング・システムとして利用可能である。

● HELP機能

IRISにはHELP機能があり，HELPファイルに登録された各コマンドの説明文や図化情報を画面に出力できる。

HELP情報も知識ベースに登録されるため，if～thenルールに従って，ツリー構造のHELPメニューを作成できる。

●ユーザー・トランスレータ

カードイメージとバイナリイメージの相互のファイル変換ユーティリティ「TRFILE」の拡張機能として，形状データや解析データをユーザー用ファイル・フォーマットに変換して登録する機能がある。この変換機能はユーザー作成ルーチンと知識ベースで構成され，用意されたサブルーチンをユーザー側がコーディング・修正・追加したり，ユーザー・メニューを登録することなどにより独自のトランスレータを短時間で作成することができる。

●パラメトリック図形

直線・円弧・スプライン曲線・面などの基本図形に対し，図形生成手順を知識ベース内に組み込むことによりパラメトリック図形を生成させることができる。これにより類似形状の入力も必要なパラメータを与えるだけで入力でき，入力の省力化を図ることができる。

カスタム化の具体的な実施例

具体的な実施例として資料提供があったキヤノンの射出成形CAEシステム「CAPLAS」をあげる(**図1**)。

CAPLASプログラムは，樹脂が金型内に充填される過程を解析する「MFLOW-SHELL」と，充填後成形品を金型から取り出すまでの冷却過程を解析する「MCOOL-SHELL」の2つの解析モジュールからなる。このシステムの標準プリ／ポスト・プロセッサとしてIRISが使用されている(**図2**)。

標準のIRISのプリ・プロセッサでは物性値などの設定は構造解析や熱解析が主体であるため，以下のような諸条件をプリ・プロセッサ内で入力させるためにさまざまな改良を試みている。

■ ANALYSISでは

●解析の選択：流動，冷却，流動＋冷却

●解析レベルの選択：簡易解析，本解析

●樹脂の選択：キヤノン独自のデータベースを使用して樹脂を選択

●解析ステップ数の指定

●樹脂温度の指定

■ GATEでは

●型温のセット

●射出方法，時間のセット

物性・荷重・拘束条件の設定メニューを専用メニューに修正し，解析・境界条件などの各要求項目は知識ベースを修正してIRISのデータに割り当てている。

プリ・プロセッサで入力された種々のデータは，解析するためプリ・トランスレータでデータ変換される。このトランスレータは「TRFILE」の拡張機能を利用し，用意されたサブルーチンをコーディングして独自のトランスレータを使用している。

カスタム化に対する方針

CAD/CAM/CAEの適用業種・業務が大きく広がった現在，ユーザーのニーズが多様化し細分化されてきた。今後こうしたユーザーのニーズを調査・分析し，さらにユーザーのノウハウを組み込めるようなソフトウエア開発をしていきたいと考えている。

また，ソフトウエア開発のみならずユーザーがIRISを効率良く利用できるよう，ユーザー教育・コンサルティング・保守に積極的に取り組んでいきたいと考えている。

問合せ先：㈱エイ・アイ・テクノロジー　営業管理課
〒101　東京都千代田区東神田2-6-2
☎03(864)6912

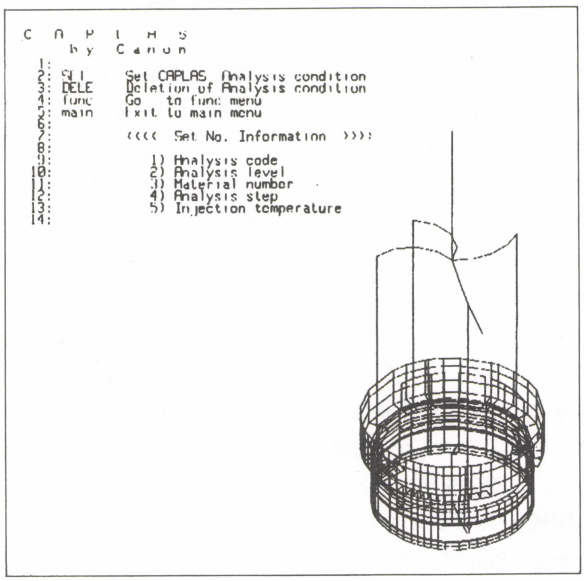


図1　CAPLAS専用の境界条件セットのメニュー

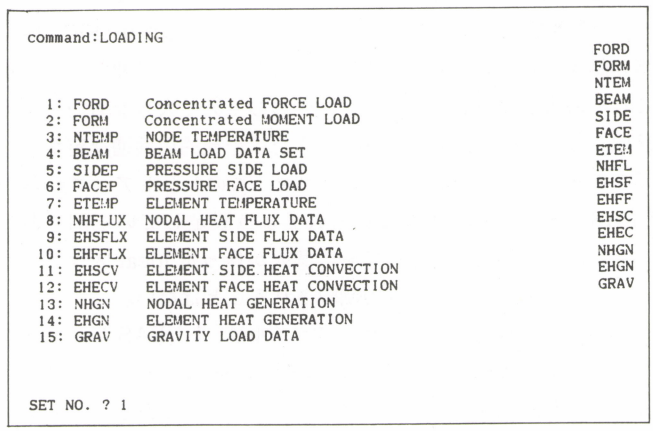


図2　IRISのメニュー

SDRC I-DEASのカスタマイズ・ツール

電通国際情報サービス

CAD/CAMシステムが提供しているカスタム化の機能

I-DEASのカスタマイズ・ツールとしては，I-DEASプログラミング言語(Ideal)を応用したI-DEAS内部でのカスタマイズと，I-DEASプロジェクト・データベース・マネージャー(Pearl)，またはSDRCユニバーサルファイルを使った他の応用プログラムとの統合化インタフェースがある。

● I-DEAS言語(Ideal)

I-DEASでは，コマンドや数値を変数で与えたり，I-DEASの出力をスカラー/ベクトル変数に代入してこれらの関数処理，演算を行ったり，IF判定，GOTO分岐を使って繰返し処理を行うなどの定型業務の自動化が可能である。また，マクロコマンドの定義，ファンクション・キーの定義，ピック可能なユーザーコマンドのメニュー出力ができ，これらを利用するとユーザーによるカスタマイズが，あたかもI-DEASのネイティブな機能であるかのようにできる。これらの一連のコマンドはIdealとよばれるプログラミング言語で書かれているが，IdealはFORTRANなどの言語について基礎知識をもっている人であれば容易に使うことができる。

● I-DEAS　PearlおよびSDRCユニバーサルファイル

Pearlは，プロジェクト・データベースとよばれるリレーショナル型のデータベースに対して，Join，Intersectなどの関係演算，作表，プロット表示を行うデータベース・マネージャーである。Pearlは，I-DEASデータベースとプロジェクト・データベース間で形状データ，FEMモデルのデータ，テスティング・データを転送する機能をもっている。つまり，ユーザー作成プログラムからもI-DEASのデータをインタフェースの容易なテーブル形式でアクセスできることを意味している。ユーザー作成プログラムからプロジェクト・データベースにアクセスする方法は，Pearlコマンドによって入出力できるテーブル形式データファイルを介するか，Pearlサブルーチン群をコールして直接アクセスするかの2通りが用意されている(**図1**)。

SDRCユニバーサルファイルは，I-DEASの応用プログラ

ム・モジュールが入出力できるカードイメージのテキスト(文字)ファイルである。このファイルは，いわばI-DEASデータベースに記録されているバイナリ・データをフォーマット化したようなもので，I-DEASから効率良く入出力することができる。このファイルのもう一つの利点は，プリンタ出力やエディタで編集できるため，インタフェース・プログラムの作成時にデバッグが容易に行えることである。

カスタム化の具体的な実施例

I-DEAS言語(Ideal)による繰返し処理や定型業務の自動化は非常に容易で，ほとんどすべてのI-DEASユーザーが利用している。同じような部品をいくつもモデル化する必要がある場合，大きさや半径をパラメータで与えるだけとしたり，FEM解析の結果としていつも最大主応力と最小主応力が必要であればこれらの表示を自動化するなど，少しの工夫で大きな省力化が達成される。また，あるグループまたは全社的に共通な定型処理があるユーザーの場合は，カスタマイズのセットアップがI-DEASの起動時に自動的に行われるよう設定している例もある。このような設定を行っても，個人が作ったIdealプログラム・ファイルへのアクセスが制限されることはない。

PearlやSDRCユニバーサルファイルは，I-DEASがサポートしているNASTRAN, ANSYS, ADAMSなどとのインタフェースでも使われているカスタマイズ・ツールである。ユーザーがインタフェース・プログラムを開発するときに必要な情報はI-DEASのマニュアルにすべて記されているが，通常I-DEASユーザーではなくプログラミングに経験のある通常のプログラマによって開発されている。当社が開発・提供しているADINAインタフェースやMoldflowインタフェースは，ユニバーサルファイルを利用している。

カスタム化に対する方針

I-DEASはI-DEAS自体のカスタマイズを許していないが，I-DEASはカスタマイズ・ツールを提供している。矛盾するように思われるかもしれないが，当社はI-DEASのカスタマイズ・ツールが現在最も良いソリューションと考えている。

●高品質と高性能の提供

I-DEASは安定した信頼性をもっている。しかし，ユーザー・サブルーチンとI-DEASがリンクされるかたちのソフトウエアであれば，障害の発生時にそれがユーザーの誤りによるものか，それともI-DEASの誤りによるものかを切り分けるのに多くの時間と費用を要し，結果として現在の安定さを実現できなかったであろう。

一方，I-DEASデータベースの仕様は，性能向上のために改良が常に加えられている。もしユーザーがI-DEASデータベースの仕様を知り，I-DEASデータベースを直接アクセスするインタフェース・プログラムを作成したとしても，I-DEASのバージョンアップのたびにユーザー作成プログラムの変更が余儀なくされ，決してユーザーの利益にはなり得ない。

●オープン・アーキテクチャ

I-DEASはカスタマイズを，I-DEASプログラムの変更としてではなく“カスタマイズ機能”として提供している。その目的とするところは，高信頼性を維持し，かつI-DEASのバージョンアップによるユーザー作成の既存カスタマイズへの影響を最小化し，I-DEASデータベースに保存されたデータへのアクセスを開放(オープン・アーキテクチャ)することである。

問合せ先：㈱電通国際情報サービス　東京営業二部
〒164　東京都中野区中野4-11-10
☎03(228)6140
開発元：Structural Dynamics Research Corp.(米国)

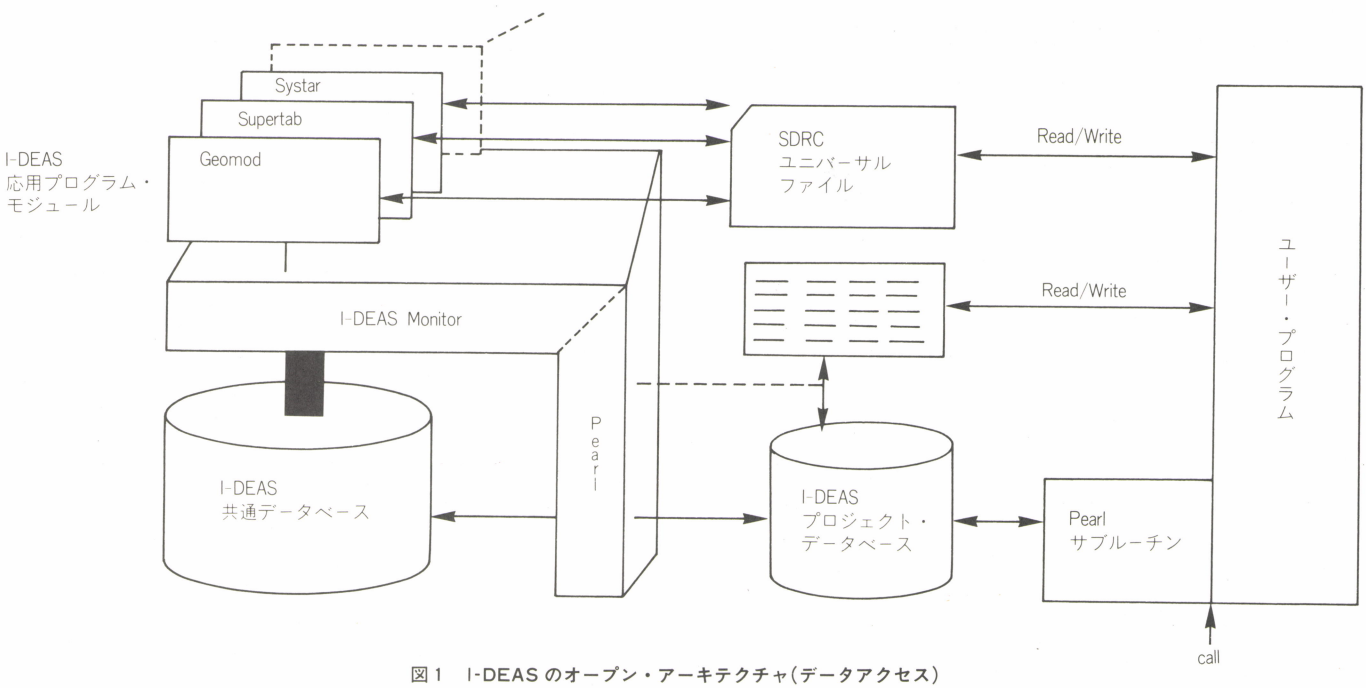


図1　I-DEASのオープン・アーキテクチャ(データアクセス)

特別寄稿

ここまできたリアリズム
——四半世紀の進展——

CGが誕生してから四半世紀，リアリズムの進歩についてSIGGRAPHを中心にその動向を探る。

中前　栄八郎*

◉ 図一部 COLOR IMAGES 参照。

はじめに

チョゴリ姿のお嬢さんのプラカードを先頭に，華やかなマーチにのって日本選手団の入場行進が続く——ソウルオリンピック開会式の一幕——。

プラカードのお嬢さん，なんと大学での専門は電子計算機，日本でコンピュータ・グラフィックス(CG)の勉強をしたいという。CGも遂にここまできたか，との想いが強い。ランドセルを背負ったこましゃくれた子が，「これがいわゆるコンピュータ・グラフィックスかー」と雑誌を広げながら話し合っているのを渋谷の本屋で見かけたのは，確か5年ほど前だったように思う。

MITの大学院生であったI.E.Sutherlandが対話型の図形表示システムSketchpadを発表して世間を驚かせたのが1963年だから，あれからすでに四半世紀が過ぎた。今日数万人を集めて催されるSIGGRAPH（Special Interest Group on GRAPHics)が，600人の参加者によって初めて開かれた1974年から数えても15年が経過したことになる。プラカードのお嬢さんはまさにCG世代の申し子であり，またNIESの追い上げの急を告げる警鐘ともとれる。今日では，テレビのコマーシャルや天気予報などCGは完全に茶の間に入り込み，“物書きにとって現在ワープロが必要不可欠な道具であるのと同じように，ごく近い将来CGが芸術家のアトリエできわめて重要な役割を占めるだろう”という記事がNews Weekのような週刊誌に載るほどまでに，先進国では広い分野にわたって実用化の時代に入った観がある。

このようにみてくると，今の時点でCGの発展の推移を概観し，今後の動向について考えることは意味のあることのように思われる。今日CGがカバーする分野はきわめて多岐にわたっている。ここでは，よりリアルな画像創成のためのCGといった立場から，主としてその表示技法のこれまでの進歩や動向について観察してみたい。内容の性質上，SIGGRAPHの論文が中心にならざるを得ない点をご容赦いただきたい。

CG創生期(1963〜1969)

前述のように，1963年にI.E.Sutherlandが対話型の2次元図形表示システムを開発すると，すぐL.G.Robertsが3次元表示法を開発した。これがコンピュータによってリアルな画像をつくり出そうとした最初のものであろう。1964年にはS.A.Coonsが自由曲面表示のための曲面要素の補間法を発表し，このような動きが大きなインパクトを世間に与え，明日にでもCGが実用化されるかのような錯覚をもたらしたのも事実である。しかし，1960年代はこのような夢と現実との大きなギャップの中で，企業，特に航空産業からの強い要望を背景にCADとしての2次元線図形表示技法の開発が進められ，今日のCADAM（ロッキード社が機械設計のための2次元製図用CADとして開発したもので，現在では広く一般企業で利用されているソフトウエア）の基礎などが構築された時期であった。

一方，3次元図形にとっては，この期間はその基礎的な図法の開発期ということができる。1966年にSutherlandが“隠面消去における10の解決すべき問題点”を指摘し[1]，1967年にはA.Appelが初めて影をもった陰影表示技法を発表した[2]。これは1980年代へ入ってからのレイ・トレーシング法（**図1**）の基礎を与えたものであるが，当時の表示装置はまだX-Yプロッタやモノクロのランダム・スキャン型CRTが主力であり，ハーフトーン表示（階調表示）をするには，**図2**に示すようなラインプリンタの重ね印字やX-Yプロッタで×印などを描くしかなく，当時の計算機の速度や記憶能力からみると負担が大き過ぎ，レイ・トレーシングが実現するまでにはこれから10年以上の歳月を要している。

このような条件のもとで，1967年にG.Romneyらが物体を三角形の集合として取り扱い，初めて透視法によるハーフトーン表示を発表し[3]，1968年にはR.A.Schumackerらが人手を介

*なかまえ　えいはちろう　　広島大学　工学部第2類(電気系)　〒724　東広島市西条町下見

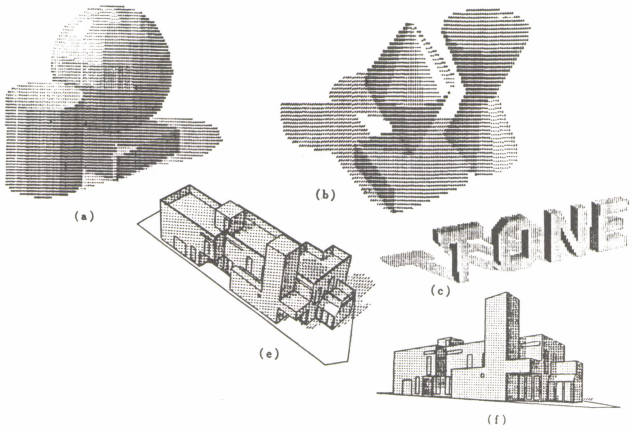


図2　X-Y プロッタとラインプリンタの重ね印字による陰影表示
（情報処理，74 年 1 月号，p.24，図 9 より）

図3　2 次元テクスチャ・マッピング
（加藤俊明氏，PIXEL 88 年 3 月号より）

してではあるが隠面消去のための優先順位法（図 1）を提案し[4]，1969 年には I.E.Warnock が画像空間領域を細分して隠面消去する方法を発表した[5]ことにもみられるように，隠面・隠線消去手法の基礎の確立にその努力が傾注された。しかし，3 次元図形に限れば，1960 年代はアメリカでも数少ないいくつかの大学・研究所で研究が行われただけで，まだ学会的な活動にはいたらなかった。このことは，1970 年代前半までの参考文献にある投稿先からみても明らかなことで，先達らの苦労を垣間みることができる。わが国では，わずかに東京大学の穂坂教授が自由曲面の発生法について研究を進め[6]，気を吐いたにすぎなかった。

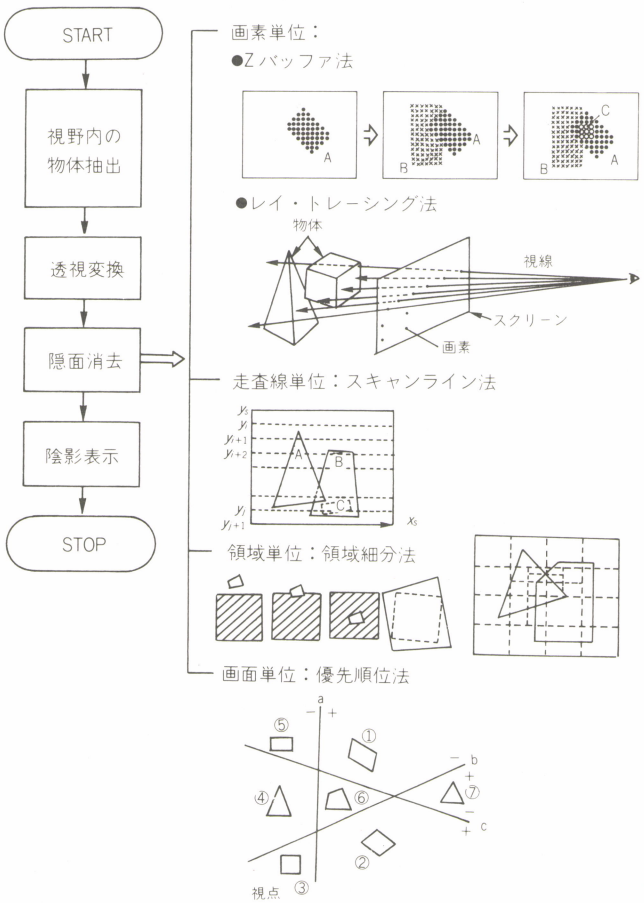


図 1　処理手順の概要

物体の陰影表示基礎技法の確立（1970〜1974）

1970 年代の前半は，ラスター・スキャン型 CRT の出現に勢いを得て，物体のよりリアルな 3 次元画像表示技法の基礎が確立された時期であったといえるのではないだろうか。ハーフトーン表示による多面体や曲面の陰影表示技法が開発され，物体の形状に関する表示技法（隠面・隠線消去）と濃淡表示技法がほぼ確立された。1972 年には人間の手の動きをアニメーション化して世間を驚かせ，また今日隆盛を極めている前述の SIGGRAPH が誕生した時期でもある。

1970 年には J.E.Bouknight が画像空間での隠面消去アルゴリズムを発表し[7]，また彼らは平行光線と点光源に対する陰影表示法を提案した。G.S.Watkins は Warnock の方法を発展させて，走査線上でのサンプルスパン・アルゴリズムを開発し[8]，スキャンライン・アルゴリズムのきっかけをつくった。さらに，1974 年には筆者らが，光源と視点のそれぞれから可視面を求めて陰影表示を行う，いわゆる 2 段階法を提案した[9]。

また，CRT のハーフトーン表示機能の向上に伴って，曲面の陰影表示が本格化した時期でもある。すなわち 1971 年には Bézier によりいわゆるベジエ曲面[10]が，また 1973 年には R.F. Riesenfeld らによって B-Spline が発表され[11]，1971 年には H. Gouraud によってスカラー補間による曲面のスムーズ・シェーディングが実現するようになった[12]。さらに，1974 年に E.E. Catmull が Z バッファ法（図 1）による曲面表示と 2 次元テクスチャ・マッピング法を発表した[13]（**図 3**）。Z バッファ法はその

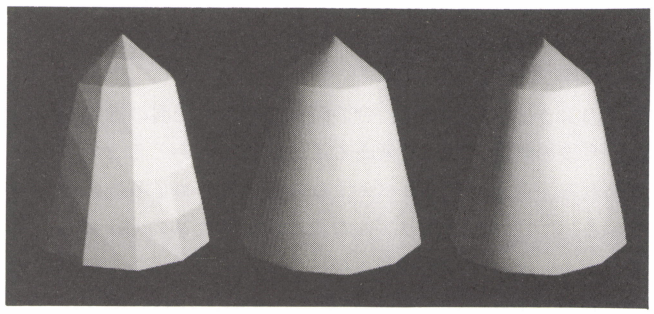
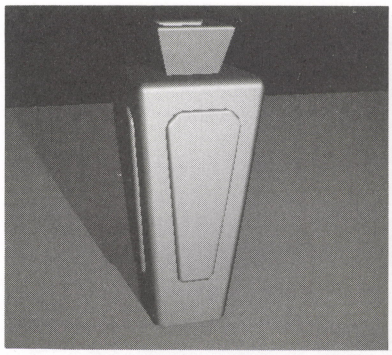
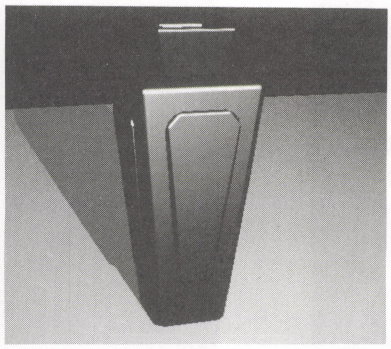


図4　曲面の表示法
左：三角パッチ，中央：輝度値の線形内挿，
右：面の法線ベクトルの線形内挿



Phong の反射モデル



Cook, Torrance の反射モデル

図5　鏡面反射モデルの比較

図6　バンプ・マッピング
（加藤俊明氏, PIXEL 88 年 3 月号より）

後，メモリの低価格化とそのアルゴリズムの簡単さのためにおおいに普及するようになった。また，光の屈折を考慮していないとはいえ，半透明表示法（1972 年）が M.E.Newell らによって示されたのもこの時期である[14]。

画質の高品質化と材質感表示法の基礎的な開発（1975〜1979）

1970 年代後半になると計算機の能力がかなり向上し，ミニコンピュータが普及するようになり，また 1975 年には 1 つのチップに演算装置を搭載した MITS 社の組立てキットが発表され，いわゆるマイクロコンピュータが世に出た年でもある。一方，記憶素子の低価格化に伴って CRT の多色化が進み，512×512 ピクセルをもったフルカラー（R，G，B 各 8 ビット，1,600 万色同時表示）のものがまだまだ高価ではあったが市販されるようになり，3 次元 CG の開発環境は少なくともアメリカでは急速に明るくなり，カラフルでリアルな映像が世間の注目を浴び始め，再びアメリカで CG の可能性が見直され始めた時期である。

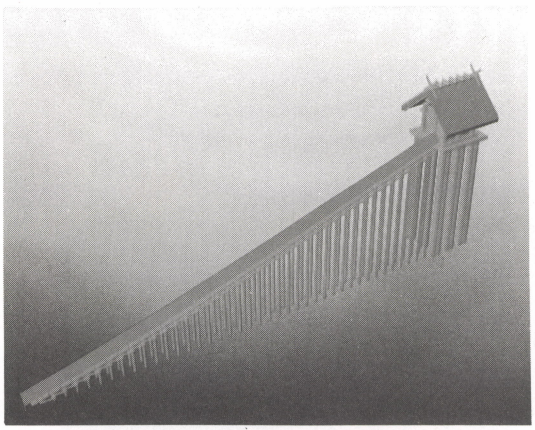
このようなハードウエア環境の進歩を背景にして，1975 年には Bui T.Phong がベクトル補間による曲面表示（**図 4**）と近似的な鏡面反射を提案した[15]（**図 5**）。さらに 1976〜78 年にかけて，J.F.Blinn が反射を含む 2 次元テクスチャ・マッピング[16]や，バンプ・マッピング[17]（**図 6**），さらに精密な鏡面反射[18]など，物体をよりリアルに表現するための技法を矢継ぎ早に発表した。

また，この期間は被照物体によって生じる影に対して関心が集まり始めた時期でもある。すなわち 1977 年に F.C.Crow が影多角形の概念を発表し[19]，1978 年には L.Williams は Z バッファ法に対する影の表示アルゴリズムを開発した[20]。

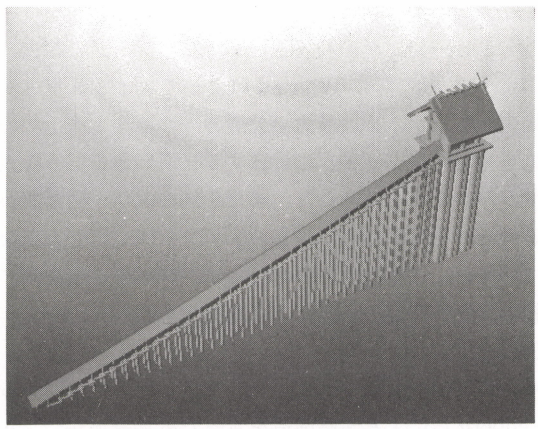
このように陰影表示技法が向上してくると，画素の粗さによるサンプリング・エラー，すなわち物体どうしの境界上の各ピクセル（画素）が明瞭に現れる，いわゆるエイリアシングが目立ち，気になるようになった（**図 7**）。この現象をソフトウエア的に除去する方法として，1978 年に面積計算法による方法が Catmull によって[21]，また 1979 年には多角形クリッピングによる方法が H.Fuchs らによって提案され[22]，1980 年代の各種アンチ・エイリアシング手法の開発の口火を切った。

3 次元 CG 表現の多様化（1980〜1985）

“The Computer Graphics Revolution” といったセンセーショナルな記事が 1981 年初頭の IEEE の Spectrum に載るまでもなく，1980 年代に入ってから CG の周辺がにわかに騒がしくなってきた。これは 1980 年に T.Whitted がレイ・トレーシング法を発表して[23]，カラー写真のようにリアルできらびやかな



エイリアシング除去



エイリアシングあり

図7　エイリアシングの除去

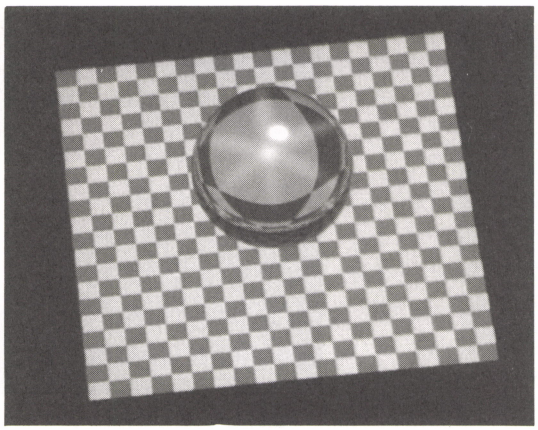


図8　異方性をもった光の屈折(横井茂樹氏，PIXEL 88 年 11 月号より)

図9　ローカル・レイ・トレーシング

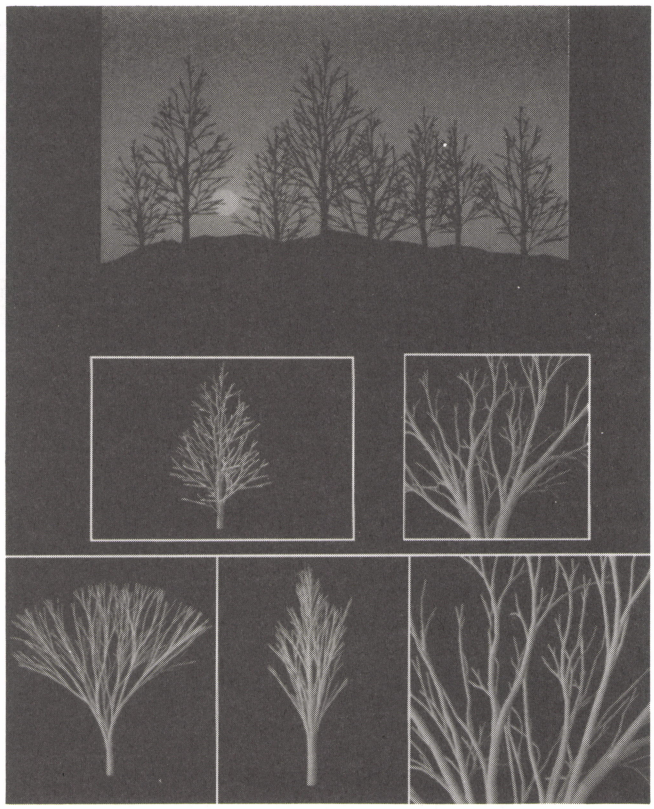


図11　成長の規則性による木の表現(千葉則茂氏ら，PIXEL 88 年 2 月号より)

画像を創出したことによるだけでなく，1983 年にゼロックス社がパーソナル化したマイクロコンピュータを発表したのをきっかけに，その後 16 ビット，32 ビットのワークステーションが次々と発表されるようになってますますコンピュータのパーソナル化が進み，コンピュータそのものが家庭にまで普及し大衆化してきたことにも起因している。この期間の CG の研究には，より複雑なものをよりリアルに，より高速に表現するための論文が数多く発表され，またいわゆる CG の研究者だけでなく放送，芸術関係を含めて各方面からの SIGGRAPH への参加者がうなぎ登りに増加した。わが国でもジャーナリズムが取り上げるようになり，SIGGRAPH へのツアーが組まれるようになったのも，このころからである。

(1)　レイ・トレーシングの高速化とエイリアシングの除去

前述のように 1980 年に Whitted によってレイ・トレーシング法が発表されたが，その最大の欠点の一つは，1 つ 1 つの画素ごとにレイを発生させ独立に計算する必要があるため膨大な計算時間がかかることであった。またそれにより，画素間が互いの関連性をもたないために生じるエイリアシングも大きな問題として残った。これらの克服のために，Whitted 自身同じ年に引き続いて物体空間の分割による高速化を試み[24]，1983 年には J.C.Cleary によってマルチプロセッサによるハードウエアの面からの高速化が提唱され[25]，一方，ソフトウエアの面からは D.P.Greenberg らによる階層構造をもった外接領域法[26]，A.S. Glassner や東京大学の國井教授らによる種々の物体分割法など[27),28]，枚挙にいとまがないほど高速化手法が競われるようになった。エイリアシングの問題についても，1984 年には J. Amanatides のコーン・レイ・トレーシング[29]，P.S.Heckbert らのビーム・レイ・トレーシング[30]が発表され，反射，屈折，透過現象の表示も含めて，よりリアルな画像表示と華麗な画像が続々と発表されるようになった（**図 8，9**）。

(2)　自然物の表示技法の開発

この期間に開発された 3 次元 CG の主な分野としては，前述のレイ・トレーシング法の他に，これまでの人工物体だけでなく山や木などに代表される自然物の表示法の着手があげられる。

すなわち，1982 年に A.Fournier や L.Carpenter は，パラメトリックなパッチに確率統計推計要素を加えた方法と，三角形パ

図10　フラクタルによる山の表現(宮田一乗氏，PIXEL 88 年 2 月号より)

図12　半透明テクスチャ・マッピングによる雲

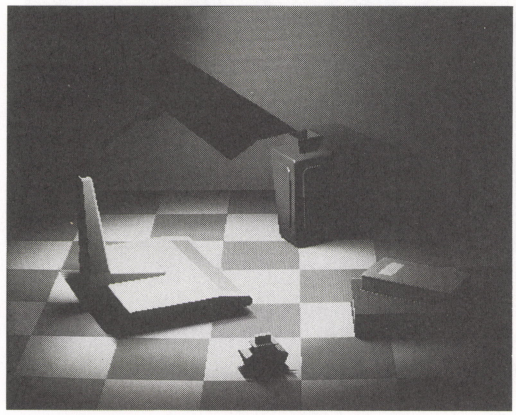


図13　線光源による半影と本影

ッチを再帰的に分割する方法によって地形を表現し[31]（**図 10**），1983 年には W.T.Reeves が推計法を用いた手続きモデルを発表し[32]，炎や爆発の様子をアニメーション化し，その後さらにこれを発展させて草や木などの発生を可能にした[33]。次いで 1984 年には國井教授らと A.R.Smith がそれぞれ木の成長規則モデル[34),35)]（**図 11**）を，G.V.Gardner が印象派モデルともいえるモネの絵画を想起させる半透明のテクスチャ・マッピング法を発表[36]（**図 12**）し，自然物体の表示法の開発意欲を触発した。

(3)　各種光源の開発

また，1980 年代の前半は各種光源による光と影のシミュレーションが注目を集め始めた時期でもある。1983 年に面光源のような大きさをもった光源による半影の表示法が筆者らによって発表され[37]，また配光特性や線光源についても開発がなされ

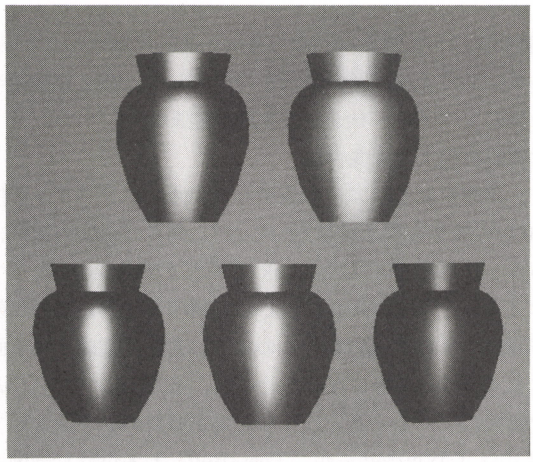
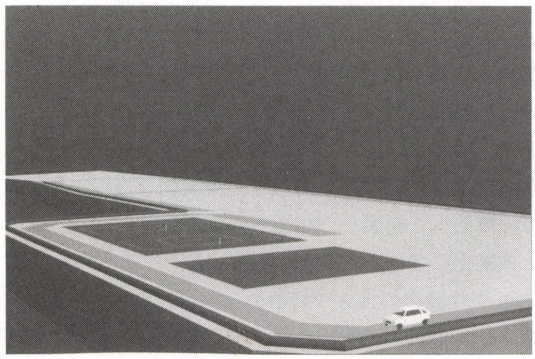
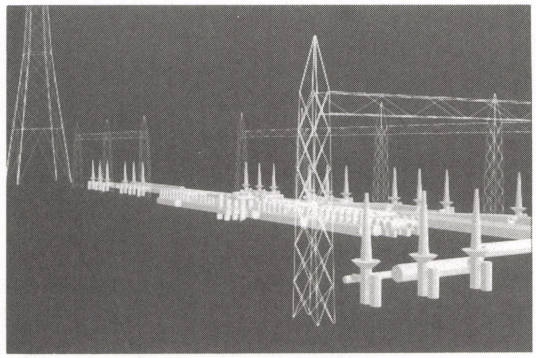


図14　金属の鏡面反射　左下から金，銀，銅，左上から鉄，アルミ

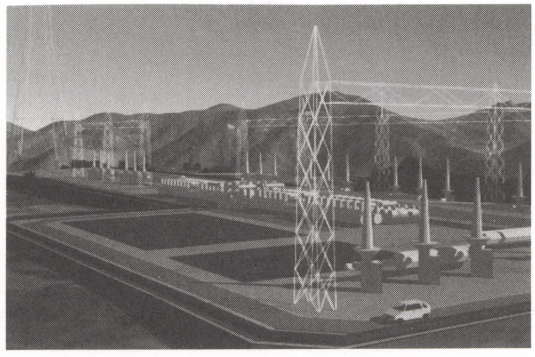


土　台



電力設備

背　景



合成後

図15　画像合成の例

◉ 図 13～15 は前ページ。

図16　モーション・ブラ
（日本ビクター㈱　総合システム研究所，柴本猛氏より）

た[38]（**図 13**）。また 1984 年には J.Amanatides がコーン・レイ・トレーシングによる半影の表示を可能にし[29]，1980 年代後半の多様な光のシミュレーションの基礎が固まった。

(4)　材質感の向上と画像合成

一方，物体をよりリアルに表示するためのレンダリング技法が 1970 年代後半に引き続き用いられた。例えば金属の反射についていうと，1982 年に R.L.Cook と K.L.Torrance によって光のスペクトルを考慮に入れたきわめて精巧な金属の鏡面反射モデルが発表された[39]（**図 14**）。このように CG で創成される画像がますます巧妙で精緻になってくると，必然的にプログラムも複雑で膨大になってくる。これに対処するための手段として，1981 年には T.Whitted らが“Test-Bed”と名付けた画像合成法を発表し[40]，1984 年には L.Carpenter がジャグフリーの合成法を開発し[41]，プログラムのモジュール化が進められた（**図 15**）。また，エイリアシングの除去法が数多く発表され，特に 2 次元のテクスチャ・マッピングの問題がほぼ解決された時期でもある。主なものとして 1980 年の C.F.R.Weiman のズーミングを許すもの[42]，R.L.Cook らによる軸対称フィルタ[43]，1982 年の A.Norton らの関数系のテクスチャに対するもの[44]，1983 年の L.Williams による Mip Map とよばれる階層構造をしたエイリアシング・フリーのテクスチャ・マッピング[45]，1984 年の F.C.Crow による高速なマッピングなどがあげられる[46]。

その他，1982 年には M.Potmesil らはこれまでのピンホールカメラ的な CG 画像に焦点深度をもたせたものを発表し[47]，1983 年にはこれをさらに発展させて時間軸のサンプリング誤差によるエイリアシングの除去を動きのブレに利用することを提案し[48]（**図 16**），よりスムーズな動きの表現法の研究の口火を切った。

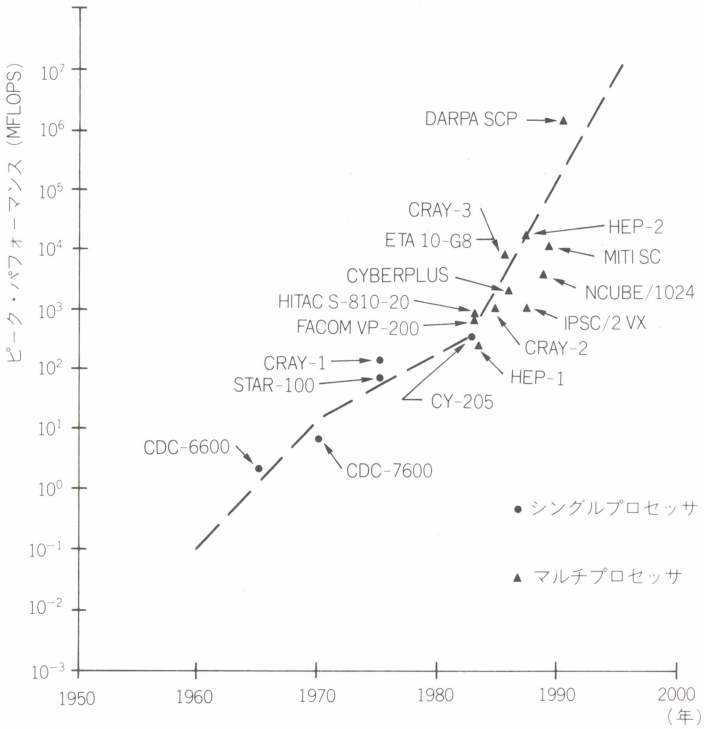


図 17　スーパー・コンピュータの速度の推移
MFLOPS：100 万回／秒のフローティング・ポイント演算

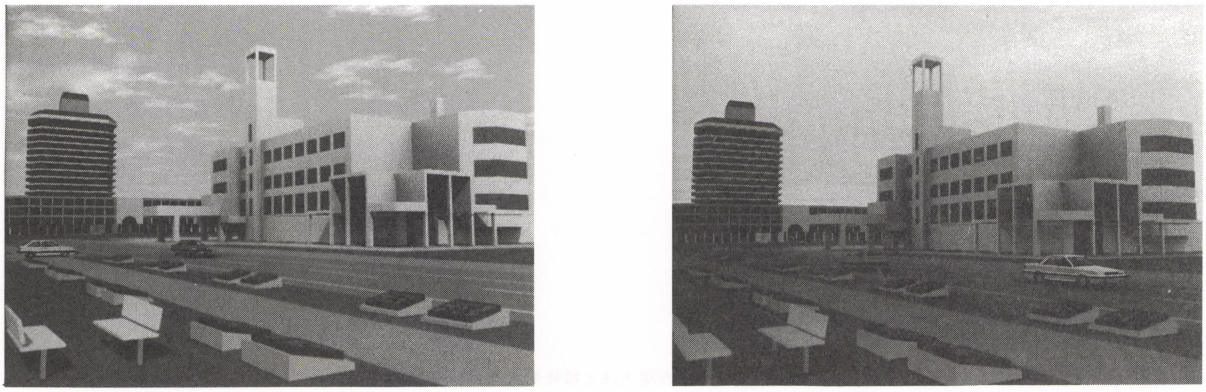
自然物理現象の表現（1985～）

1980 年代前半を山や樹木など自然物体の表現法の開発が人々の注目を集めた時代とすると，その後半は自然界の物理現象表現法の開発に関心が集まっている時代ということができようか。**図 17** に示すような計算機の高速化・高機能化に伴って，これまでの人工的な光源だけではなく，光の相互反射，天空光，光芒といった自然現象としての光と影の問題，また自然界の物理現象，すなわち波，雲や炎といったこれまでの個体とは異なる液体や気体の動きを含めた表現，また個体についてもこれまでの固定した形状の表示だけではなくその力学的な運動や変形問題が取り扱われるようになり，人間や動物の動きについても，写真や実物の動きのコピーとしての CG ではなく，人体の運動機能，筋肉の動きに基づいたより科学的な裏付けのあるリアルなアニメーションの試みをはじめ，さらに鳥や魚の群れの動きなどエコロジー（生態学）をも取り入れた広い意味での自然現象の表現技法の開発へと広がりをみせてきた。

少し大げさにいえば，これからの CG はもはや単なる絵画的な表示技術から一歩踏み出して，視覚的な森羅万象を表示するための技法へと発展しつつあるかにみえる。

(1)　本格化してきたアニメーション

1980 年代後半の大きな特徴の一つは，本格的な 3 次元アニメーションへの試みが開始され，多くの人々の注目を浴びていることである。もちろん，これまでにも多くのアニメーションが制作されてきたが，これらは例えば，車の動きを加速度，軌道などをあらかじめ設定して内挿法を用いて行ったり，人や動物の動きも実物の動きのコピーや簡単な関数による近似であったりした。言い換えると，物体の動きについては個体の人為的な移動であって，アニメーターの手作業に負うところが多く，弾性体，非弾性体の外力による運動や変形，衝突現象，布のような柔軟な物体の動きや波などの液体の変形などについてはほとんど手がつけられていなかった。これらの動きを忠実に描くことはまさにこれら物理現象のシミュレーション結果の表示そのものであり，一般的には 3 次元空間内での非線形現象の過渡解析に他ならない。したがって，解析そのものに多大の時間と記憶容量を必要とするものであり，物理的な諸条件をどこまで近



晴天時　　曇天時

図 20　天空光を計算した屋外風景

似的に取り扱うかという問題に帰着する。その近似の度合いは表示目的——例えば，純粋な物理現象の解明の手段としてや，フライトシミュレータの表示への適用，コマーシャル・フィルムやファイン・アートへの利用など——によって異なる。それにしても，このような問題が取り扱えるようになった大きな要因は，ハードウエアの進歩による計算速度の高速化，メモリチップの大容量化，それに伴う計算機の低価格化とパーソナル化や，ネットワーク化によるソフトウエアの開発環境が急速に良くなってきたことによるところが大きい。

1986 年は，W.T.Reeves らと D.R.Peachey が，波の動き，特に海岸線での波の複雑な変化の表現に期を同じくして成功し[49),50)]，また J.Weil によって初めてカーテンのような柔らかい布のはためきがカテナリー曲線を用いて表現された[51)]年である（**図 18**）。引き続き A.Barr らは，弾性理論をゴムや絨毯や旗などに適用して，これらの動きを外力に対する時間的な変形問題として取り扱った。すなわち，これらの運動を表す微分方程式を解くことによってリアルなアニメーションを可能にした[52)]。さらに，1988 年には，これらの柔軟構造体の衝突や外力による変形のアニメーションを発表し[53)]，また鎖のように剛体のつながったものの振動を表示する物理モデルを矢継ぎ早に発表した[54)]。1988 年はこのような物理モデルのオンパレードともいえるほど，種々の物理モデルが発表された年でもある。主なものだけでも K.Fleischer らの非弾性体の変形，破砕を取り扱ったもの[55)]，同じく自動車や椅子などの非弾性体の運動を取り扱った J.K.Hahn のもの[56)]，また剛体どうしの衝突問題を domino に適用してアニメーション化した J.Wilhelms らのもの[57)]，スプリングの振動作用を利用して軟体動物（蛇やみみず）のリアルな表現を行った G.S.P.Miller のもの[58)]など，枚挙にいとまがない。

一方，人体のアニメーションについても，これまでのキーフレームの内挿によるものから，より力学的要素を加えた現実の動きに忠実な手法の模索が開始された。すなわち，1985 年には M.Girard らは人間の足の動きについて手続き的な手法を提案し，足首と地面との相互作用を考慮したものを開発し[59)]，1987 年には F.Cohen らがこれまでのキーフレーム・アニメーションを許容する“運動の拘束条件”と，動作に必要な力と加速度に対する時々刻々の状態を表す動作関数，および特定の動きを実行するのに必要な力を決めるための“逆問題”を組み合わせた方法を提案し[60)]，各関節の調和のとれた動きを開発してダンスや体操などの人体のより自然な動きの創成に意欲をみせている。また，人体の顔の表情はきわめて微妙で最も表現の難しいものの一つであるが，K.Waters は顔の筋肉モデルを提案し[61)]，この面での一つの突破口を開いた。

この他，1987 年に発表された C.W.Reynolds による鳥や魚などの群れとしてのアニメーションは生態学的な手法によるものであり[62)]，今後の広範囲にわたる分野でのアニメーションの可能性を示唆している。

このように 1980 年代後半は，計算機の高速化とソフトウエアの開発が相まって本格的でリアルな 3 次元アニメーションの可能性を切り開きつつあるということができる。

(2)　より精密な環境光表現

1980 年代後半は，光と影の面からも写真のようなリアルな自然の光環境の表示を可能にしつつある。すなわち，前述のように，1980 年代前半の光源に関する基礎的な開発をもとにより微妙な光と影の問題が議論されるようになった。すなわち，これまで直射光の当たらない影の部分は，いわゆる環境光とよばれる適当な一定の明るさが与えられていた。1985 年には，これに取って代わるものとして物体表面を完全拡散面と考え，それらの相互反射による間接光の効果を表す方法が，D.P.Greenberg らと筆者によって発表された[63),64)]（**図 19**）。これは熱の表面放射と

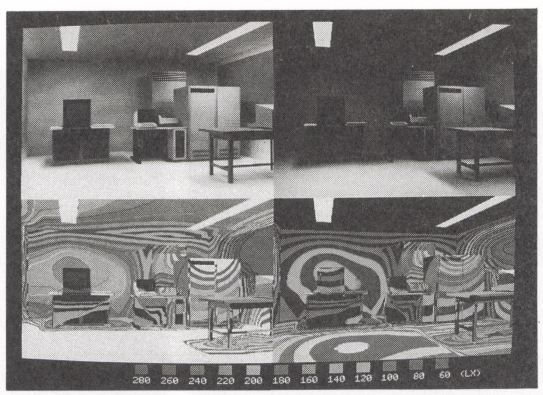


図 19　光の相互反射　Radiosity
左：相互反射を考慮した表示とその照度分布
右：直接照度のみの表示とその照度分布

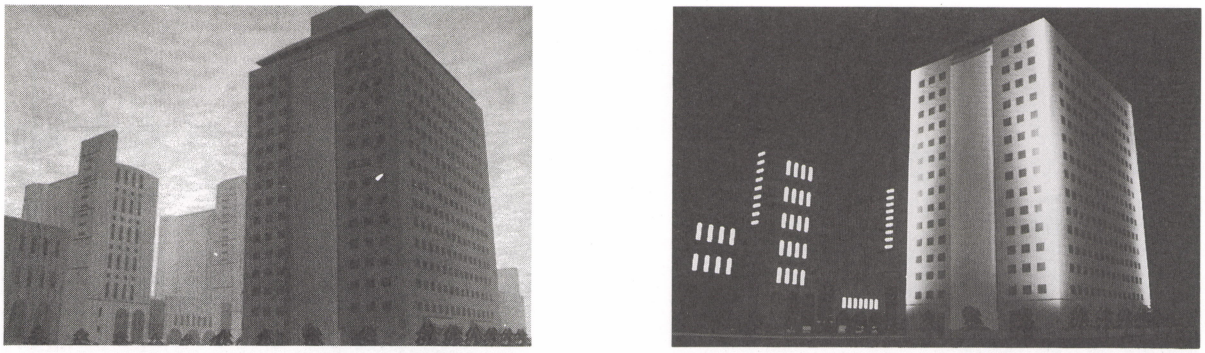
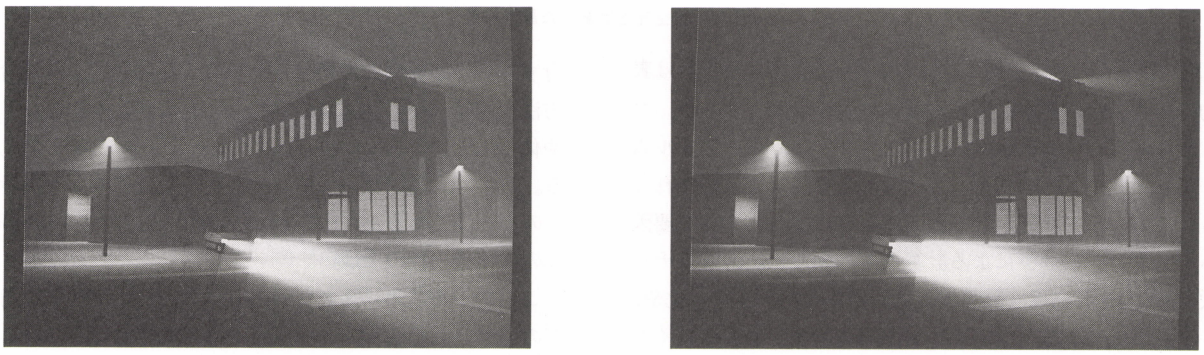


図 21　指数関数を用いた遠近法と投光器で照らされたビル



フェイズ・ファンクションなし　　フェイズ・ファンクションあり

図 22　ヘッドライトで照らされた霧の表現

同じ考え方が利用できることから，この効果は Radiosity とよばれるが，計算時間がかかるのが難点であった。Greenberg らは引き続き鏡面反射を含むものや高速化についての研究を進め[65)~67)]，一見写真と間違えるようなきわめて写実的な画像を創出して人々を驚かせた。また，1988 年の SIGGRAPH の展示会ではワークステーション上で Radiosity を表現するものが現れ，実用化へのテンポの速さが人目を引いた。ここで扱われる相互反射効果は，主として屋内または屋内への光の射込みに対して有効に活用できる。これに対して，屋外における環境光ともいえる天空光を筆者らが開発し[68)]，晴天時や曇天時の建物の微妙な見え方の変化の表現が可能になった（**図 20**, 前ページ）。

霞効果による遠近感の表示法としては，これまでは単に距離が遠くなるに従って指数関数的に色彩のコントラストを落とす方法が用いられていた（**図 21**）。これに対して，1986 年に N.L. Max が大気中の微粒子による光の放射・散乱・吸収効果を考慮し，自然光による雲間からの光束の表示法を発表し[69)]，1987 年には室内への陽射しやヘッドライト効果の表示などが K.E. Torrance らや筆者などにより実現し[70),71)]（**図 22, 23**　さらに R.V.Klassen によって太陽光の大気分子やエアゾルによる屈折・散乱・吸収効果における天空の色の表現，すなわち夕焼けなどの自然現象の表示が可能になった[72)]（**図 24**）。

(3)　精緻なモデルと高速処理

上述のようなリアルな動きと光環境表現が可能になるとともに，より精巧で精緻なモデルの開発が試みられるようになった。その一つは，1985 年に D.R.Peachey によって行われた 3 次元テクスチャであり[73)]，これによって木や大理石の任意のカット

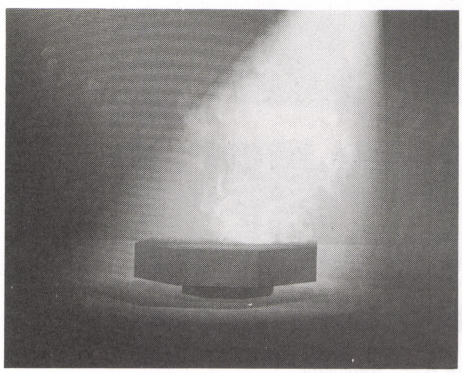


図 23　煙と光束の表現

図 24　空の色の表現

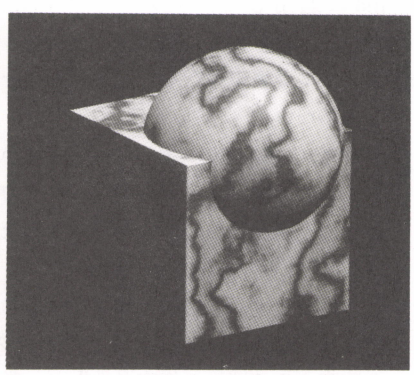


図 25　ソリッド・マッピングによる大理石の表現
（横井，岡田氏，PIXEL 88 年 8 月号より）

断面の木目や肌理表現が可能になった(**図 25**)。植物，樹木についてもさらに細かい表現とアニメーション化のために，フラクタルを用いた分岐モデルが P.E.Oppenheimer によって[74]，また光日特性などを加えた植物の生長過程を表現するものが P. Prusinkiewicz らによって[75]，より精密な木の法則性を取り入れたモデルが P.D.Reffye[76]らによって次々と発展しつつある(**図 26**)。このような個々の表示対象物のよりリアルな表現法の開発とは別に，無数といってよい自然物の複雑な形状を高速に表現するためにこれらを微細な小片に分解し，階層リスト構造とすることによって高速に処理しようとする A.H.Barr らの試み[77]は注目に値する。彼らは，例えば 4,000 億個もの小片から構成される広大な草原の光景の表示に成功している。

このようなソフトウエアの面からの高速化は，特にレイ・トレーシングに関して 1980 年代前半に引き続き種々試みられているが，1980 年代後半の高速化の特徴は，むしろハードウエアによる高速化が研究段階としてではなく実用化に根差した開発として進められていることである。その主なものとして 1985 年の富士通の井上氏らによるセル・アレイ・プロセッサによるソリット・モデルの高速表示システム[78]がある。また H.Fuchs らによる "Pixel-Planes" とよばれているフレーム・バッファを構成するアレイ構造の論理メモリチップ[79]は，単に画像の記憶だけでなくピクセル・レベルでの領域の決定，可視計算，着色が可能で，さらに凸多面体集合の高速，陰影，テクスチャ，半透明的表示を行うエンジンである。また，1987 年に J.G.Torborg が発表したパラレル・プロセッサは単一プロセッサと同じようにプログラムできるので，マルチプロセッサの利点を生かしながら簡単にソフトウエアの開発を行うことができる[80]。1988 年に入ると，シリコングラフィックスがパイプライン・プロセッサ，ベクトル・プロセッサ，アレイ・プロセッサを駆使した並列化によって 1 秒間に Gouraud の shading を施した 10 万個の四辺からなるポリゴンの表示可能なアーキテクチャを発表し[81]，ステラ・コンピュータは同じく 15 万個の三角形／秒を実現できる製品を発表し[82]，いよいよリアルタイムのスーパー・ワークステーションの時代ともいえる時期を迎えた感を呈し始めた。

おわりに

以上，リアルな画像の創成技法の観点からこの 25 年間の CG の歩みを概観した。

それでは，今後 CG はどのような発展を遂げるのであろうか。筆者の頭で予測するにはあまりにその進歩が激しく，そのマーケットも大きな広がりをみせつつある。前述したように，1980 年代は「コンピュータ・グラフィック大革命」というにふさわしいほどその表現能力を向上させ，きわめて今日的な話題性に富んだ分野として世間からかなり脚光を浴びているが，道具としてはまだまだ未熟で使い勝手の悪いものであり，その活用には専門知識が要求され，また高価でもある。

しかし，CG の開発者と利用者の人口は着実に増加しており，いよいよ本格的な実用化の時代へ向かってその門戸を開きつつある。1990 年代には CG の利用が日常茶飯事化し，高度情報化社会において欠くことのできない道具として使われるようになるであろう。

参考文献

1) I.E.Sutherland："Ten Unsolved Problems in Computer Graphics", Datamation,Vol.12,No.5, p.22,1966
2) A.Appel："Some Techniques for Shading Machine Renderings of Solids",Proc. AFIPSJSCC,Vol.32, pp.37-45,1968
3) C.Wylie,G.Romney, D.Evans, A.Erdahl："Half-tone Perspective Drawings by Computer", AFIPS Conf. Proc., Vol.31,pp.49-54,1967
4) R.A.Schumacker et al.："Study for Applying Computer Generated Images to Visual Simulation", AFHRT-TR-69-14,1968
5) J.E.Warnock："A Hidden-Surface Algorithm for Computer Generated Half-tone Pictures", Univ. Utah Comput. Sci. Dept. TR4-15,1969
6) 穂坂衛：「曲線，曲面の合成および平滑化理論」，情報処理，Vol.10,No.3,pp.121-131,1969
7) W.J.Bouknight, K.C.Kelly："An Algorithm for Producing Half-Tone Computer Graphics Presentations with Shadows and Movable Light Sources", Proc. AFIPSJSCC, Vol.36,pp.1-10,1970
8) G.S.Watkins："A Real-Time Visible Surface Algorithm", Univ. Utah Comput.Sci.Dept.,UTEC-CSc-70-101,1970
9) T.Nishita, E.Nakamae："An Algorithm for Half-Tone Representation of Three-Dimensional Objects", Information Processing in Japan, Vol.14,pp.93-99,1974
10) P.Bézier："Example of an Existing System in the Motor Industry；UNISURF System", Proc.Roy.Sec.Lond., A321, pp.207-218,1971
11) R.F.Riesenfeld："Application of B-Spline Approximations to Geometric Problems of Computer-aided Design",Univ.Utah Comput.Sci. Dept., UTEC-CSc-73-126,1973.3
12) H.Gouraud："Continuous Shading of Curved Surfaces", IEEE Trans.Computers,Vol.C-20,No.6,pp.623-629,1971.6
13) E.E.Catmull："A Subdivision Algorithm for Computer Display of Curved Surfaces",Ph.D.thesis, Univ. of Utah,1974
14) M.E.Newell,R.G.Newell, T.L.Sancha："A New Approach to the Shaded Picture Problem", Proc.ACM, Nat. Conf., p.443,1972
15) Bui T.Phong："Illumination for Computer Generated Pictures", Comm.ACM,Vol.18,No.6,pp.311-317,1975.6
16) J.F.Blinn, M.E.Newell："Texture and Reflection in Computer Generated Images", Comm.ACM,Vol.19,No.10,pp.542-547,1976.10
17) J.F.Blinn："Simulation of Wrinkled Surfaces", Computer Graphics, Vol.12,No.3,pp.286-292,1978.8
18) J.F.Blinn："Models of Light Reflection for Computer Synthesized Pictures",Computer Graphics,Vol.11,No.2,pp.192-198,1977.7
19) F.C.Crow："Shadow Algorithms for Computer Graphics",Computer Graphics, Vol.11,No.2,pp.242-248,1977.7
20) L.Williams："Casting Curved Shadows on Curved Surfaces", Computer Graphics, Vol.12,No.3,pp.270-274,1978.8
21) E.E.Catmull："A Hidden-Surface Algorithm with Anti-Aliasing", Computer Graphics, Vol.12,No.3,pp.6-10,1978.8
22) J.Barros, H.Fuchs："Generating Smooth 2-D Monocolor Line Drawings on Video Display", Computer Graphics, Vol.13,No.2,pp.260-267,1979.8
23) T.Whitted："An Improved Illumination Model for Shaded Display", Comm.ACM, Vol.23,No.6,pp.343-349,1980.6
24) S.M.Rubin, T.Whitted："A 3-Dimensional Representation for Fast Rendering of Complex Scenes", Computer Graphics, Vol.14,No.3,pp.110-116,1980.7
25) J.G.Cleary et al.："Multiprocessor Ray Tracing", Dept.of Computer Science, Univ. of Calgary, Research Report, No.83/128/7,1983
26) H.Weghorst, G.Hooper, D.P.Greenberg："Improved Computational Methods for Ray Tracing", ACM Trans., Vol.3,No.1, pp.52-69,1984.1
27) A.S.Glassner："Space Subdivision for Fast Ray Tracing", IEEE CG&A, Vol.4,No.10,pp.15-22,1984.10
28) K.Yamaguchi, T.L.Kunii, K.Fujimura："Octree-Related Data

Structures and Algorithms", IEEE CG&A, Vol.4,No.1,pp.53-59,1984
29) J.Amanatides : "Ray Tracing with Cones", Computer Graphics, Vol.18,No.3,pp.129-136,1984.7
30) P.Heckbert, P.Hanrahan : "Beam Tracing Polygonal Objects", Computer Graphics, Vol.18,No.3,pp.119-128,1984.7
31) A.Fournier,D.Fussell, L.Carpenter : "Computer Rendering of Stochastic Models", Comm.ACM,Vol.25,No.6,pp.371-384,1982.6
32) W.T.Reeves : "Particle Systems—A Technique for Modelling a Class of Fuzzy Objects", Computer Graphics, Vol.17,No.3,pp.359-376, 1983.7
33) W.T.Reeves, R.Blau : "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems", Computer Graphics,Vol.19,No.3,pp.313-322,1985.7
34) M.Aono, T.L.Kunii : "Botanical Tree Image Generation", IEEE CG&A,Vol.4,No.5,pp.10-34,1984.5
35) A.R.Smith : "Plants, Fractals and Formal Languages", Computer Graphics, Vol.18,No.3,pp.1-10,1984.7
36) G.Y.Gardner : "Visual Simulation of Clouds", Computer Graphics, Vol.19,No.3,pp.297-303,1985.7
37) T.Nishita,E.Nakamae : "Half-Tone Representation of 3-D Objects Illuminated by Area or Polyhedron Sources", Proc.COMPSAC 83, IEEE Computer Society, pp.237-242,1983.11.7-11
38) T.Nishita, I.Okamura, E.Nakamae : "Shading Models for Point and Linear Sources",ACM Trans. Graphics,Vol.4,No.2,pp.124-146, 1985.4
39) R.L.Cook,K.L.Torrance : "A Reflectance Model for Computer Graphics",ACM Trans.Graphics, Vol.1,No.1,pp.7-24,1982.1
40) T.Whitted, D.M.Weimer : "A Software Test-Bed for the Development of 3-D Raster Graphics Systems", Computer Graphics, Vol.15, No.3,pp.271-277,1981.8
41) L.Carpenter : "The A-buffer, an Antialiased Hidden Surface Method", Computer Graphics, Vol.18,No.3,pp.103-108,1984.7
42) C.F.R.Weiman : "Continuous Anti-Aliasing Rotation and Zoom of Raster Images", Computer Graphics, Vol.14,No.3,pp.286-293,1980.7
43) E.A.Peibush, M.Levoy, R.L.Cook : "Synthetic Texturing Using Digital Filters", Computer Graphics, Vol.14,No.3,pp.294-301,1980.7
44) A.Norton,A.P.Rockwood, P.T.Skolmoski : "A Method of Antialiasing Textured Surfaces by Bandwidth Limiting in Object Space", Computer Graphics, Vol.16,No.3,pp.1-8,1982.7
45) L.Williams : "Pyramidal Parametrics", Computer Graphics, Vol. 17,No.3,pp.1-11,1983.7
46) F.C.Crow : "Summed-Area Tables for Texture Mapping", Computer Graphics, Vol.18,No.3,pp.207-212,1984.7
47) M.Potmesil, I.Chakravarty : "Synthetic Image Generation with a Lens and Aperture Camera Model", ACM Trans. Graphics, Vol.1,No. 2,pp.85-101,1982.4
48) M.Potmesil : "Modeling Motion Blur in Computer-Generated Images", Computer Graphics,Vol.17,No.3,pp.389-399,1983.7
49) W.T.Reeves : "A Simple Model of Ocean Waves",Computer Graphics, Vol.20,No.4,pp.75-84,1986.8
50) D.R.Peachey : "Modeling Waves and Surf", Computer Graphics, Vol.20,No.4,pp.65-74,1986.8
51) J.Weil : "The Synthesis of Cloth Objects", Computer Graphics,Vol. 20,No.4,pp.49-54,1986.8
52) A.H.Barr, D.Terzopoulos, J.Platt, K.Fleischer : "Elastically Deformable Models", Computer Graphics, Vol.21,No.4,pp.205-214,1987.7
53) A.H.Barr, J.C.Platt : "Constraint Methods for Flexible Models", Computer Graphics, Vol.22,No.4,pp.279-288,1988.8
54) A.H.Barr, R.Barzel : "A Modeling System Based on Dynamic Constraints", Computer Graphics, Vol.22,No.4,pp.179-188,1988.8
55) D.Terzopoulos, K.Fleischer : "Modeling Inelastic Deformation : Viscoelasticity, Plasticity, Fracture", Computer Graphics, Vol.22,No. 4,pp.269-278,1988.8
56) J.K.Hahn : "Realistic Animation of Rigid Bodies", Computer Graphics, Vol.22,No.4,pp.299-308,1988.8
57) M.Moore,J.Wilhelms : "Collision Detection and Response for Computer Animation", Computer Graphics, Vol.22,No.4,pp.289-298,1988.8
58) G.S.P.Miller : "The Motion Dynamics of Snakes and Worms", Computer Graphics, Vol.22,No.4,pp.169-178,1988.8
59) M.Girard,A.A.Maciejewski : "Computational Modeling for the Computer Animation of Legged Figures", Computer Graphics, Vol.19, No.3,pp.263-270,1985
60) P.M.Isaacs,M.F.Cohen : "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics", Computer Graphics, Vol.21,No.4,pp.215-224,1987.7
61) K.Waters : "A Muscle Model for Animating Three-Dimensional Facial Expression", Computer Graphics, Vol.21,No.4,pp.17-24,1987.7
62) C.W.Reynolds : "Flocks, Herds and Schools : A Distributed Behavioral Model", Computer Graphics, Vol.21,No.4,pp.25-34,1987.7
63) M.F.Cohen,D.P.Greenberg : "A Radiosity Solution for Complex Environments", Computer Graphics, Vol.19,No.3,pp.31-40,1985.7
64) T.Nishita, E.Nakamae : "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection", Computer Graphics, Vol.19,No.3,pp.23-30,1985.7
65) D.S.Immel, M.F.Cohen, D.P.Greenberg : "A Radiosity Method for Non-Diffuse Environments", Computer Graphics, Vol.20,No.4,pp.133-142,1986.8
66) J.R.Wallace, M.F.Cohen, D.P.Greenberg : "A Two-Pass Solution to the Rendering Equation : A Synthesis of Ray Tracing and Radiosity Methods", Computer Graphics, Vol.21,No.4,pp.311-320,1987. 7
67) M.F.Cohen,S.E.Chen, J.R.Wallace, D.P.Greenberg : "A Progressive Refinement Approach to Fast Radiosity Image Generation", Computer Graphics, Vol.22,No.4,pp.75-84,1988.8
68) T.Nishita, E.Nakamae : "Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light", Computer Graphics, Vol.20,No.4,pp.125-132,1986.8
69) N.L.Max : "Atmospheric Illumination and Shadows", Computer Graphics, Vol.20,No.4,pp.117-124,1986.8
70) H.E.Rushmeier, K.E.Torrance : "The Zonal Method for Calculating Light Intensities in The Presence of a Participating Medium", Computer Graphics, Vol.21,No.4,pp.293-302,1987.7
71) T.Nishita, Y.Miyawaki, E.Nakamae : "A Shading Model for Atmospheric Scattering Considering Luminous", Computer Graphics, Vol.21,No.4,pp.303-310,1987.7
72) R.V. Klassen : "Modeling the Effect of the Atmosphere on Light", ACM Trans. Graphics, Vol.6,No.3,pp.215-237,1987.7
73) D.R.Peachey : "Solid Texturing of Complex Surfaces", Computer Graphics, Vol.19,No.3,pp.279-286,1985.7
74) P.E.Oppenheimer : "Real Time Design and Animation of Fractal Plants and Trees", Computer Graphics, Vol.20,No.4, pp.55-64,1986.8
75) P.Prusinkiewicz, A.Lindenmayer, J.Hanan : "Developmental Models of Herbaceous Plants for Computer Imagery Purposes", Computer Graphics, Vol.22,No.4,pp.141-150,1988.8
76) P.D.Reffye, C.Edelin, J.Francon, M.Jaeger, C.Puech : "Plant Models Faithful to Botanical Structure and Development", Computer Graphics, Vol.22,No.4,pp.151-158,1988.8
77) J.M.Snyder, A.H.Barr : "Ray Tracing Complex Models Containing Surface Tessellations", Computer Graphics, Vol.21,No.4,pp.119-128,1987.7
78) H.Sato, M.Ishii, K.Inoue et al. : "Fast Image Generation of Constructive Solid Geometry Using a Cellular Array Processor", Computer Graphics, Vol.19,No.3,pp.95-102, 1985.7
79) H.Fuchs, J.Goldfeather, J.P. Hultquist et al. : "Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes", Computer Graphics, Vol.19,No.3,pp.111-120,1985.7
80) J.G.Torborg : "A Parallel Processor Architecture for Graphics Arithmetic Operation", Computer Graphics, Vol.21,No.4,pp.197-204, 1987.7
81) K.Akeley, T.Jermoluk : "High- Performance Polygon Rendering", Computer Graphics, Vol.22,No.4,pp.239-246,1988.8
82) B.Apgar, B.Bersack, A.Mammen : "A Display System for the Stellar™ Graphics Supercomputer Model GS1000™", Computer Graphics, Vol.22,No.4,pp.255-262,1988.8

テクニカル・イラストの変遷とCAD化への可能性

デジタル情報伝達の社会において，伝達効果，圧縮性，迅速性，加工性の高く新しい媒体が模索されている。この役割を果たす媒体がベクトル化された立体図である。

西岡　和紀*

はじめに

国内における立体図の本格利用の元年ともいえる昭和36年ころ，今はすでに生産中止となった国産初の中型旅客機YS-11によって正式な画法による立体図の採用が実施され，大量の立体図作成のためのシステムが考案されていく。

この揺籃期，今は亡き富田素忠先生の果たした役割の偉大さにわれわれは感謝しなければならない。先生は，戦後の航空機産業の受難時，元立川米空軍基地での航空機設計，補給カタログ作成システムを実地取得され，その貴重な経験をYS-11のサービス・ドキュメント業務に100%活用し，今日の立体図作成の基礎を確立した。特に，このプロジェクト終了時(昭和41年)に発刊された「テクニカル・イラストレーションの基礎画法」によって，学問的な体系化，実践に基づく制作手法，仕様を決定した。さらに，軸測投影法による基礎作図理論を設定し，等測(等角)投影図法によって効率化の道を開いた。それまで絵の世界で非効率に作成された立体図作成手段に，おおいなるショックを与えたのである。

一つの基礎画法が確立されることによって，周辺の用具，用紙，制作手段，教育の全分野で開発テーマが明確になり，急速に業界としてのかたちが整備されていくのである。

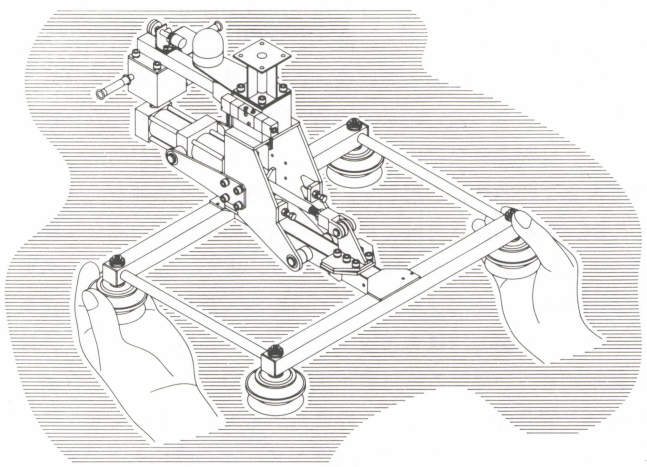


図1　CADによる作図例

以下，現在にいたるまで20余年の種々の変遷を振り返り，将来の方向を模索し，今，なぜCAD化が必要なのか? 改めて考えてみる。

当社では，すでに40台近いCADが稼働している。昭和59年に1号機導入以来，どのようにCAD化を進めたのか? CAD化への動機は何であったのか? CADの効果はあったのか? CAD化によって何が変化したのかを，ここにオープンにして読者諸氏と一緒に考えてみたいと思う(**図1**)。

需要の背景

YS-11の需要にみられるように，立体図利用の大半は補修部品の補給カタログ指図用としてである。本来は設計図面をそのまま利用すればよいわけであるが，設計図に対する機密漏洩の問題，仕向地(海外諸国)，業者との言葉の違いなど，部品調達上の不備点を解消するために立体図が採用されている。特に広大な地域間のトラブルは，生産性に大きな問題を引き起こす。

このような立体図を基本媒体としたパーツ・カタログの発想は，第二次世界大戦にさかのぼる。広域な戦場を維持し，破壊された武器の補修，消耗品の調達を迅速に実行するため，補給兵站部隊が多用したといわれる。日本においても，戦国時代の絵巻物を見ると戦時の武器補給調達が絵によって行われていることに気がつく。これらの時代においては，補給部品調達の正確度，タイミングがそのまま生命を左右する出来事であろうことが容易に予測できる。さらに，アメリカにおいてこのようなパーツ・カタログが普及を続けた背景は，第二次世界大戦以降の中東問題，東南アジア，朝鮮半島の社会情勢をみても想像できるし，現在もまさしく臨戦状況と変わりない情景さえ推測できる。

さらにアメリカの航空機産業は，自由圏諸国への戦闘機輸出の役割も大きい。また，ジャンボ旅客機を頂点とした種々の民間航空機の開発，生産も世界の需要の大半を賄っており，航空機のもつ特有な定期整備性あるいはサービス拠点の多面性などの理由により，立体図を利用したパーツ・カタログなどの需要

* にしおか　かずのり　　名立エンジニアリング㈱　〒467　愛知県名古屋市瑞穂区桜見町2-12

が最も高く，その利用体系は産業界に大きな影響を与えている。大量の人命を預かり，部品欠陥を起因とする万一の事故も引き起こせない航空機業界のサービス体制を先達として，人命を預かり移動型の形態をもっている自動車・トラックなどの自動車製造工業においても，全国各地の販売店における部品サービスネットの情報源として立体図化された部品帳を完備している。さらに類似形の業種として建設機械業界も，自動車業界ほど大規模ではないが立体図利用体系に移行している。

以上，代表的な利用例をみるかぎり，立体図の利用は，大量生産品，地域制限性のない移動型製品，人命預託型製品の製造業が最初は渋々体制作りを始め，その効果が発生すると大規模なサービスシステムの伝達媒体として立体図を重視し，今日にいたっている。ちなみに，国内における自動車業界の年間補給部品販売額は8,000億円に達する。そして，多用化する商品群，ライフサイクルの短い商品群，輸出依存企業のいずれもサービス体制の欠如が販売戦略に与える影響力に気づき，航空機・自動車業界の形態を基本にしながら昭和50年代後半に立体図を媒体として，種々のサービス資料(特にパーツ・カタログ)の作成をサービスシステムの根本に位置付けし，システム整備を行い，未完成ながら発展の方向を示し，現在にいたっている。

このように，立体図利用の背景は日本独自の発想ではなく先進国よりの押付け型であり，さらに人命重視型製品の採用により後押しされ，普及している。現在，全製造業における採用率は60%を超えている。そして，その利用部門はサービス部門によって90%以上占められている。

用具の変遷

設計図面を作成するための用具は，明治時代より英式・独式と若干の相違があったが，円を描くためのコンパス，寸法採り，転写のためのデバイダーなどを代表に今日まで基本形状を変えずに利用されている。製図機においても，T定規と三角定規を利用した方法が昭和30年代はもちろん，昭和40年代の前半まで一般的であった。昭和40年代前期より普及し始めた製図機械もアーム式の製図機が主流を占め，昭和50年代初めのトラック式製図機の出現までその地位を占めていた。しかし，現在アーム式製図機を見ることは皆無に近い。製図機においてすらこのような状況の中で，昭和40年代初期の立体図作成の用具は，国産の製図機メーカーより購入することは不可能であった。このような環境の中で，富田先生の提唱された軸測投影法はタイムリーな動きとしてとらえられ，輸入業者のアメリカ製用具の紹介，国産製図機メーカーにおける国産化の方向へと進むのである。

■楕円定規

立体図作成には短軸・長軸の比率の違う楕円が大半を占め，製図用のコンパスでは作図が不可能である。そのため，各種の角度をもったテンプレートが必須な道具となる。当時，塩化ビニール製の薄板加工技術が未確立であり，必要の生じた作成者は，自らボール紙やプラスチック板を切り抜いた手作り品を用いて制作していた。

このような手作り用具において極小楕円定規を作ることは不可能であり，大きな楕円テンプレートを利用した立体図のサイズはA1サイズから畳1枚の大きさにまで及んだといわれる。そのためコスト対応が不可能となり，一時的にデッサンの得意な人たちが絵として立体図を作成する画法無視の方向をたどり，現在においてもその技術が継承されている。ただ，このような動向は急速に増えた輸入品のテンプレートにより解決の方向を見いだしていくのである。さらに，国産の製図機メーカーが製造を開始し，ついに安価で良質なテンプレートの製造に成功するのである。

昭和45年ころであったと記憶している。テンプレートについては第2次ともいえる立体図の高需要期にタイミング良く量産が可能になったことにより，急速に立体図を生業とする業者が乱立を始めたのである。

■T定規およびドラフタ

一方，立体図には120°の角度で交わる X, Y, Z の三面を構成する基軸線が必要である。この水平面に対して左右に30°の角度をもつ線分を作成することも，正投影の世界と大きく異なる部分である。

T定規を利用していた昭和30年代後半においては，30°の三角定規によってこの基軸線を左右へ傾けながら数多くの線分を描いた。これが徐々にアーム式ドラフタの普及とともに立体図作成に活用されていく。ドラフタの角度分割は5°の角度刻みであり，30°の基軸線作成には随分と労力を費やすものであった。そのため，ドラフタに3本のスケールをもった専用製図機も開発されたが，操作勝手の悪さや画面のエリアに占めるスケールの配分などに問題があり，ほとんど実用化されずに現在実機を見ることもない。最終的には立体図作成用の専用製図機の開発は行われず，製図機の不自由を承知したうえで利用し，現在にいたっているのである。

■トレース用ニードルペン

また，製図機械と楕円テンプレートなどで描かれた立体図は種々の作図補助線が図中に数多くあり，元図のまま次工程(印刷)へ移行できない。そのため，インキングトレース(墨入れ)を施し，印刷版下としての形態を整える必要があった。

昭和30年代後半において，トレースをするための用具はカラス口だけであった。カラス口は，T定規，三角定規の端面に沿って利用する形状になっている。この用具を楕円形状に合わせて使うことになった。このような時代がほぼ10年近くも続いた。これは熟練者だけの世界であり，大量の需要が発生した時期に大きな問題を引き起こす。今から考えると単純な問題ではあるが，複雑な曲線を描くには万年筆形の用具なら簡単に解決できることに気づかなかったのであり，また，国内において万年筆形のニードルペンが存在していなかったのである。特定業者がアメリカ製の特殊専用ペンを購入して利用していたが，他業者が知ることになるのは数年後のことであった。このため，新規参入業者は，トレース工程において大きな遅れをとることになった。その半面，製図機械輸入メーカーがドイツなどで市販されている万年筆形トレースペンを目新しい商品として試験輸入を始めたのもこのころである。難渋していただけに，1社の採用と同時にその効果が認識され，急速に普及していくのであ

る。その後，国内メーカーも生産を始めたが，ドイツ製の製品を上回る物はいまだに現れていない。ペンによるトレース手法が確立されて20年になるが，現在カラス口を使う人を見ることはほとんどない。

万年筆形のトレースペンは当初，0.3～0.5mmが主流であったため，ペン先の太さに合わせた元図の作成が要求され，必然的に用紙サイズが大きくなり，コストが高くなる要因であったが，10年前より0.1～0.2mm程度のペン製造技術が安定したことと，楕円テンプレートの極小化が進んだ結果，A1よりA2，A2からA3，A4判へと作成サイズが変化し，生産性，作業性に大きな貢献を果たした。ペン1個の精度が立体図作成の歴史を大きく変えたのである。また，CAD化時代になり，ペン・プロッタを利用すると，再度このペンのお世話になり，ニードル式ペンの選択が正しかったことが証明された。

■用 紙

さらに作成用紙の変遷も見逃せない。

絵として描かれていた時代には，カラス口の補助として筆を使ったトリミング，あるいは白ペイントを利用した修正などを行うため，厚手のケント紙が利用されていた。これが万年筆形のペンを利用する時代になると作成工程にも変化が起こり，トレーサーとしての職種が独立してくるのである。それに伴い，元図を下に置き，透過性の良いトレーシング・ペーパーへ写すようになる。この時点では薄手の紙でよかったが，新人の参加で修正の工程も無視できなくなり，破れず，透過性が良く，トレース性の高い用紙の要請がでてきたのである。窮すれば通ずの例えのごとく，このときも地図・測量用図面の作成原図として伸縮性と湿度に対する対策を講じた特殊紙が開発されていたのである。昭和48年であったと記憶している。表裏に従来のトレーシング・ペーパーを内部に塩化ビニールを配し，サンドイッチ状に成形した紙である。以降，現在にいたるまでほぼ全業者で利用されている。最近では，さらに改善された和紙を利用した製品も利用されている。CAD化に移行した現在も，紙の高速な送りスピードと，紙の圧着する力に耐える用紙として多用されている。

■まとめ

以上，25年に及ぶ用具・用紙の変遷が，コンピュータ化を図るうえで多くのヒントを与えるポイントをもっていることに気がつく。楕円テンプレート，等測基軸線は，コンピュータのソフトウエアですべてカバーされる。さらに，トレース用のペンはペン・プロッタのペンによって代替され，トレース用の特殊紙はペン・プロッタのピンチローラの圧力に耐えることができる。

コンピュータおよびCADソフト，周辺の機器は，システムとして上記のすべての要素を盛り込める。すなわち，道具として25年間で蓄積された技術，その他のノウハウのすべての肩代わりが可能なのである。

作図技術者(テクニカル・イラストレーター)の道程

四半世紀に及ぶ年月を経て，作図技術者(テクニカル・イラストレーター)は今，大きな岐路に立っている。

手描きによる従来手法の継続か？ 転業か？ さらに廃業を余儀なくされていくのか？ あるいはコンピュータ時代への対応のため，CAD時代へ突き進むのか？ 生死を賭けた問題が山積みしている。

なぜこのような状況まで放置していたのか，どうすれば解決でき今後の四半世紀を運営できるのかを，これまでの25年間を振り返りながら考えてみる。

■創業期(昭和36～43年)

立体図作成が本格化したのは，昭和36～37年ころのYS-11の生産時であったことは前項でも述べた。

立体図の生産実績が皆無に等しい環境へ，いきなり膨大な需要が飛び込んだのである。YS-11は国をあげてのナショナル・プロジェクトであり，戦後の民間機製造の第1号である。航空機の製造図は，従来の経験では解釈できない難解なものであった。機体，油圧，空調配管，エレキシステム，各種のシステムが大量の図面で構成され，立体図面の作成以前の図面解読とシステム解釈に問題が発生した。当然，機械設計経験者が立体図作成の候補者として選ばれ，故富田素忠先生の指導下で立体図の基礎画法から学習を始めたのである。潤沢な予算と納期の設定下，恵まれたスタートを切り，本格的な立体図生産が始まったのである。富田先生は，正しい理論体系を作り，数多くの技術者を育成し，今日の業界の基礎を作られた。

YS-11のプロジェクトは数年間継続され，正統な理論体系と高度な業務に従事した技術者らはプロジェクト終了と同時に次々と独立し，未開拓の分野である産業機械，自動車業界，精密機械業界へと立体図の普及に全力投球し，需要の基礎を形成した。

この時代，立体図は新技術としてもてはやされ，各メーカーはこぞって立体図の採用を行い，技術者不足もあり長い間，高価格な取引と超多忙が続いた。

YS-11に始まり他業界への啓蒙，普及を経て，またまた航空機生産のピークに出合うのである。三菱重工業のMU-2シリーズ，新明和工業のUS-1，富士重工業のT-1など，当時の技術者は最も恵まれた環境下で第1次のブームを謳歌したのである。

このように，創業期は航空機生産のピークによってサポートされたものであり，真のノウハウをもたなかった素人集団が教育，育成された時期といえる。

また，航空機のような特殊なオーダーであったからこそ，高価格，長納期という恵まれた条件がそろったのであり，このときの甘い環境が業界の形成に役立ったことは事実であるが，この後に起きる撤退劇をみるかぎり，真の技術力革新を怠った遠因になっているのも真実である。

■中興期(昭和44～54年)

需要の活発な数年が過ぎ航空機のプロジェクト終了とともに終止状態に入り，蓄積された立体図作成のパワーが一般産業機械へ振り向けられることになる。当時の工作機械をはじめとする産業機械メーカーは低価格な商品を海外向け1本にしぼり，外貨確保に必死になっていた。汎用旋盤などを1社で月産500～600台と生産し，輸出していたのである。

当然，補修業務の基礎が確立されているアメリカでは，購入

にあたっての条件としてパーツ・カタログの常備は必須であったが，メーカーでは立体図利用の体験もなく，設計図をそのまま使うことで逃げ切ったのである。まだ海外におけるサービス業務の重要性を認識していなかったといえる。

ところが，国内有力メーカーが立体図活用のパーツ・カタログを整備すると，現地ディーラーが一気に他メーカーに同等品を要求し始め，産業機械メーカーにも立体図のオーダーが発生した。ただ，価格・納期面では航空機の50%以下で提示され，品質・納期・価格のすべての面で大幅な改善を実施しないと対応できなくなり，大きな技術革新が要求されたのである。

10〜15年の開発サイクルをもつ航空機オーダーが終了した以上，輸出型商品にその活路を見いだすしかないわけであり，避けて通れない関門であった。製図法の確立，用具の開発・改善，新人の教育手段など，昭和44年から昭和48年ころにかけて種々の工夫が各社で実施された。その結果，昭和48年初期には低コスト化の対応力がつき，現在の原形に近い作業工程が完成した。

従来，1人の技術者が全工程を担当し，トレースによって製品の良否が定まっていた。すなわち，鉛筆作成の工程を自分のわかる範囲まで省略し，ほぼ完成した状態で構図の変更なども盛り込みながらインキングトレースを施し，完成させていた。もともと元図作成の技術は，図面の解読・組立状態の把握，さらに立体図法の取得と幅広いものであり，1〜2年の経験でコスト的に採算のとれる製品作りは不可能である。そこへ異質なトレース作業を組み込んだところに，これまでの作業工程に問題があった。高いコスト，長い納期が許される時代では済んだことではあるが，しかし，20年を経た今日でも大半の立体図作成会社はこの工程を採っている。後述するが，この非生産的手法が今後の大きな障害になっていくのである。

この旧来手法を改善するには工程の分離が最も近道である。

比較的簡単に訓練できるトレース業務を分離して2次工程へ回す，1次工程である元図作成者は製図板に向かって製図に専念するのである。1次工程は鉛筆で作図し，客に検図の依頼をし，その間にトレーサーがトレースを先行する。非常に簡単なことであるが，この時期よりテクニカル・イラストレーターの職質が大きく変わっていくのである。テクニカル・イラストレーターと呼称せず，立体図制作技術者，立体図トレーサーと分業体制が確立されていき，トレーサーの技術レベルが高まるにつけ元図作成の用紙サイズまで影響を及ぼしてくる。A0判からA1判，さらにA2判からA3判と乾式複写機の発達に伴うように変化した。

ちょうどこのころ，中東問題に端を発したオイルショック，ニクソン・ショック，ドルショックに遭遇し，国内の輸出産業が大打撃を受けるのである。

輸出企業が大半のユーザーである立体図業者は，相次ぐ受注のキャンセルを受ける。キャンセルを避けても大幅な値引きと，大変なピンチを迎えた。四苦八苦してコスト低減の努力をし，体制を整えた直後である。さらに各社，創業10年以上，あるいは5〜6年と経て，10〜50人程度の人員を擁していただけに，作業量確保不良の業者，コスト対応力のない業者を中心に，半数以上の会社が転廃業を余儀なくされたのである。創業期よりおよそ15年経過していた。このときに立体図の将来を見切った人たちの大半は2度と業界に復帰していない。立体図のもっている魅力に取りつかれた会社，あるいは個々の技術者らが，この困難な時代をかろうじて生き残ったのである。現在，会社を主宰している40歳代の経営者，あるいは大手の40歳代の技術者のみが知る，小さな歴史である。

このような産業界全体を揺さぶる現象の中で，立体図の作成価格がどんどん低下していき，さらに多くの改善を施す必要がでてきた。この時点でのコスト改善は前述した分業制の工程削減しかなかったのである。すなわち，鉛筆描きの状態(1次工程)で製品化し，品質の劣化に目を閉じ，ひたすらに市況の回復を待つことであった。しかし，このときの不況は長く，これだけの改善でも対応が不可能になり，遂に部品データベース化をもくろむことになる。繰返し利用する部品をすべてワッペンにして貼っていくのである。その後，普及の著しい乾式複写機で複製し，製品化した。このとき繰り返し受難した出来事が現在のCAD化にすべて役立っていることを思うと，いかなる困難な時期においても最善を尽くすことの大切さを感じている。

昭和40年代後期より昭和50年代初期の数年間は業界の大きな受難期であった。その半面，業者によっては，印刷部門・デザイン部門・設計部門・テクニカル・ライティングの分野へと，立体図作成の技術を生かした進出も活発に行われ，後日発生する第2次需要期を控え大量の技術者を温存した動向も見逃せない。現在の立体図専門業者を体系別に分けるならば，ほとんど上記の部門を主体あるいは補助的にもちながら運営されている。すべてがこのときを教訓にしていると思われるし，現在100%立体図専門業者は全国的にみても1%にも満たない。

当時の立体図作成の従事人口は全国で200名を割っていた。しかし，現在の業界の底辺はこの時期に作られたものであり，今後の指導者層もこのときの若者の中から現れることは確実である。

■発展期(昭和54〜60年)

長い不況もようやく回復期に入り，産業界において革新的な技術改革が実施され，新しい船出をするための諸条件をすべて満たし，意気上がる年始めであったと記憶している。

例えば，工作機械業界においてはNC制御技術が確立され，従来の手動型の機械から半自動・全自動化の方向を歩みだし，産業革命に等しい総入替えに近い需要を作り出すのもこの時期である。高級化，省力化への移行である。

同様に，オーディオ機器メーカーにおいてもビデオカメラ・ビデオデッキを普及させ，市民の生活パターンを一新するのである。また，宅地造成，高速道路工事も実施され，建設機械が次から次へとモデルチェンジされていった。さらに，民間航空機の開発，防衛庁向けの戦闘機の開発も相次ぎ，まさしく立体図の2次ブームが訪れたのである。

皮肉ではあるが，立体図業界ではこのブーム直前には身を小さくし，最小の規模でまさしく細々と営業していた。需要は発生したが，人手不足の状況で好況期を迎えたのである。不況期に独立した人々，30〜50人の技術者を維持していた会社もある

が，大半は5〜6名の集団であり，15人の組織なら大手ともいわれた。創業期のブームはすべてに環境が恵まれていたが，今回の需要期は納期面で非常に苦しい立場に追い込まれ，価格より期限を争うため人手なしでは消化できない状況になり，各社は次々と外注政策をとる一方，新人の確保・教育・訓練に追い込まれる。

東京で立体図の教育を専門に行う学校が出現したのもこのころであり，在京の業者が自ら講師となり卒業生を自社内に就職させるような現象も数多くみられた。さらに，全国の業者などの連絡協議会，業務の融通を図るための協会も発足し，強化されていく。また，会員相互間の技術協力向上のため，教育教材が数多く企画され，急速に立体図作成者の育成が行われた。さらに，国家検定の職種にも採用され，通信教育などで技術を取得したアマチュア・レベルの技術者が大量に有資格者として業界に参加するのである。

このようにして短期養成された人たちは，基礎工学的知識・機械・装置類の基本構造の知識は教育されず，立体図の基本作図技術のみを身につけた新しい世代のイラストレーターであった。

昭和50年代の中期には立体図の本格利用も20年を迎え，技術レベルに対する評価は随分と高くなっており，ユーザーであるメーカー各社の担当者は自社用の仕様を作成し，すべてを業者に委ねることはなくなっていた。

メーカーにおいて，立体図を利用したパーツ・カタログはすでに業務体制に組み込まれ，市民権を得ていたのである。当然，オーダーの発生と同時に業者間の競合が日常茶飯事になり，図の品質・価格・納期面すべてにチェックの目が注がれた。数年前までは品質低下を許容したうえでの生産指示であったものが，景気回復とともに再び高品質・短納期・低価格への要望が高まった。

絵から図面への転換でもある。「ただ図が並んでいればよい，それ以上の期待はしないから…」が一変したのである。この背景には数年前より始まった，全産業レベルでの技術革新の動きを見逃せない。新技術によって構築された種々の装置は，高精度の維持と高生産性を要求されていたのである。トラブル解決は迅速に，補給部品の調達も正確に速くである。

立体図を媒体としたパーツ・カタログに，機能的な役割と部品調達時の精度が要求されてきた。このメーカーの要望は立体図業界に大きな変化を起こし始めた。

技術力が高く，製品の解析力に強い，部品データまで処理できる会社へオーダーが流れ始めたのであり，総合技術力の問われる業態へと変化していく。前にも述べたように，立体図作成会社の若手社員は立体図の基礎画法のみを学んだ製図士であり，複雑な組立てを理解する訓練も不足し，経験もない。社内教育の指導力の良否が，即，生産性の低下を招く事態になったのである。経営者，ベテランの技術者の職人芸からチームワークによる組織力を問われたのである。この時点より，業界では極端に人員増を控え，少数精鋭主義をとり始めた。個人プレーに卓越していても組織力による業務遂行が苦手であり，当事者も訓練されていなかったのである。

一方，作業の増え続けるメーカーではこの消極策に待ち切れず，技術指導を提供する代わり人材の派遣を要請してくるようになり，折からの派遣社員ブームに乗り，技能検定合格者が有資格者として大量に派遣会社に採用されていく。その半面，30人程度の技術者を擁する業者では，現状維持の努力と組織力重視策をとり，発展・成長を目指したのである。各社が三者三様の形態をとりながら数年推移し，今度はCAD化問題に遭遇していくのである。

CAD化への基本的な考え方

■コンピュータでイラストは描けるのか?

手描きにおける作業道具は，X-Y-Z軸を作るためのスケール，楕円を描くための楕円テンプレート，鉛筆，それに楕円分度器である。

まず，これらの道具をすべてコンピュータが肩代わりできるかどうかを考えてみる。楕円は真円を短軸の比率変換で自由な角度に作れ，テンプレートは一切不要になる。次は基軸線。CADソフトは平面図を作成する専用機であり，角度の指定を与えるだけであらゆる角度に簡単に対応する。鉛筆は，マウス，ライトペンのスイッチ操作でカバーできる。最後は楕円分度器。楕円分度器を作るとき，計算式を用いたはずである。その計算式をコンピュータにインプットすればワークツールとして用意でき，数値入力のみで利用できる。

現在，市場に出ているすべてのCADソフトが立体図作成を可能にしている。

■トレースはできるのか? 文字・番号は記入できるのか?

ペン・プロッタを利用すれば，ペンサイズ0.1〜0.5mm，用紙もA4からA2判，いずれも自由に選択できる。通常の分解図は10分程度で出力できる。製品の出来具合は，単線処理比較では20年の経験者もかなわない。さらに，レーザービーム・プリンタ，サーマル・プリンタ，静電プリンタなど，ペン・プロッタより数段性能の高い機械が安価に出回っている。特に，夜間の出力業務(無人化)は魅力のある動向である。これらの出力機は近い将来，文字データとともに編集や出力が可能になり，版下なしの状態で印刷まで実行できるようになる。

一度描いたものを再度転写する，今の時代この発想は許されない。トレース業務は姿を消してしまうだろう。当社では昭和63年12月に95%までトレースを廃止した。

CADソフトは独自の図形文字をもっている。書体に規制されなければサイズ・太さ・傾斜文字など自由に出力できる。また，出力機によっては自由に文字フォントを選択でき，写植機と同等の文字も使えるのである。

■コンピュータでどの程度のイラストが描けるのか?

立体図は空想図ではない。すべて設計図を基本にしている。手描きにおいても，熟練技術者は瞬時に頭の中で平面図より立体感をシミュレートする。その後できるだけ効率良く，少ない線分で作図を実行する。

コンピュータで作図する場合でも，立体感を思考するのは技術者の頭であり，目であることには変わりはない。いわゆる道具としての違いのみである。道具としての違いは前述した通り，

コンピュータには多くの可能性が潜んでいることである。立体図を描くための専用機を開発するならば道具として考えればよいが，CADシステムを搭載したコンピュータは設計図を描くための専用機である。すなわち，CADによって作成された平面図のデジタル情報をどのように活用するかで作成のプロセスが一変するのである。

コンピュータにより，立体図の作成を目指すときメーカーが，なぜCADシステムを活用するのか予測・研究する必要がある。メーカーは物を作ったり金属を加工するのが仕事である。当然，CADデータを基本としたCAMデータを最終的に欲している。この欲望をもっているかぎり，メーカーのCAD化投資はとどまることはない。当然，近い将来CAD化された図面のデジタル情報が図面出図量の大半を占めることになる。それだけに，これからの立体図作成はこのCADデータを応用した作成システムを構築することが最善で最適なのである。

設計図面は正確な製造情報を与えるため，情報を欠如させることができない。その製造用データを最小限選択し，次から次へと立体変換させていくのである。ある部分はフェース単位，あるいはソリッドの単位でよい。素早く擬似立体図の姿へともっていくのである。この後は技術者の頭の仕事となり，コンピュータと対等に対話式で完成まで導くのである。コンピュータによる作図において，ワークツールは改善を怠らず，基礎データの利用体系を守ればどのように複雑な部品でも作図可能である。4年の実績をもとに結論を述べると，手描き時間以下で新人が複雑な単品図を日夜作成している。

■立体図の編集・レイアウトの問題は?

パーツ・カタログに利用される立体図作成工程で，最も技術力を必要とするのが編集・レイアウトである。単品レベルの作図では，難易度の高い部品を除くかぎり経験の差，能力の差は生じない。これはコンピュータ作図においても同様の結果がでている。

レイアウト，編集時の作業工程を大別すると次のようになる。

① フォーマットの決定(用紙サイズの決定)：ユーザー別にフォーマット作成
② 組立手順の把握：CAD化とともに計画グループへ移管
③ 単品作図およびグループ設定：CADグループ
④ 基本構図のプラン：CAD化とともに計画グループへ移管
⑤ 作成スケールの決定：CADグループ
⑥ 部品アイテム番号・タイトルの記入：CADグループ
⑦ リスト・表作成：CAD化とともに計画グループ，DPグループへ移管

これまで1人の技術者がこの工程を1人で賄ってきたのである。1枚のイラストを10時間以内で作成するには8〜10年の経験が必要である。

コンピュータの画面において，画像は自由に拡大・縮小，反転，移動，複写，消去などができることは当然といえ，大量の試行錯誤の繰返しは時間の浪費である。画面を数時間止めて，組立手順を思案することも問題が多い。特にCADは，今も，これからも，経験の少ない若人が担当することとなる。

トレース工程をCAD化によって吸収した以上，全工程を

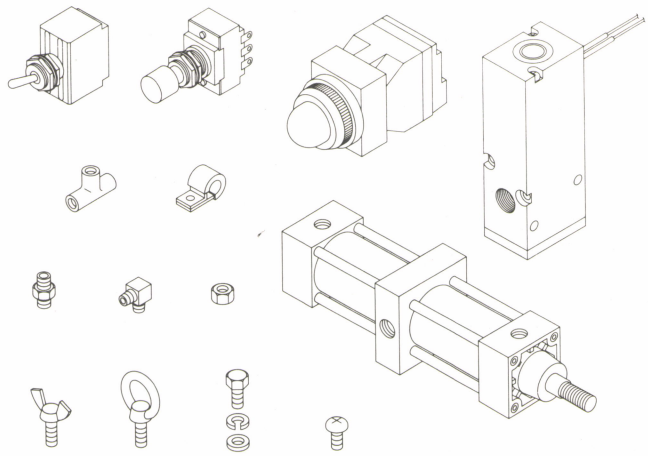


図2　立体図形のデータベース

CADグループへ委ねることは無理がある。20年近い業歴をもつ会社には手描き作業のベテランが健在している。彼らがCAD化へ移行する過程で重要な役割を果たすことができる。過去の経験を生かし，CADグループへフリーハンドで描かれた計画図を受け渡すのである。不足している技術を経験者がカバーし，CADグループの初期の立上がりを補助していく。これにより，作成スケールの決定，グループ設定などが容易になり，CADの特徴をフルに発揮させ，6カ月程度の訓練で手描き技術者の5〜6年程度の力をもつことが可能になる。

CADソフトおよびコンピュータは，立体図作成上，最大の難関である，レイアウト・構成を軽くジャンプさせてくれる。この問題が解決されるかぎり，コンピュータ導入に失敗はないと確信できる。さらに，手描き世代とコンピュータ世代とが互いに融合でき，新しいシステム展開が可能になるのである。

■コンピュータ化による部品のデータベース化

手描き時代に立体図のコスト対応のため，すでにボルトなどの部品をワッペン化したことを述べた。

コンピュータ化による最大の特徴は，フロッピディスクまたはハードディスクにコンピュータにより作成されたデータを保管・収納し，再使用できることである。手描き作品は常にオリジナルであり，再使用は複写工程を経ることになり，使用回数に比例して品質が劣化していく。さらに，縮小・拡大後の修正・改訂は不可能である。

パーツ・カタログに代表される立体図作成において，ボルトなどのファスナ類，Oリングやベアリングやオイルシールなどの機械機能部品，配管関係の継手類，スイッチ・センサーなどの電気パーツ，コネクタやレセプタクルなどの電子パーツ，さらに油・空圧シリンダ，油・空圧制御機器，これらの部品は1枚の立体図の中に常に10〜20点の発生があり，50〜100枚で構成されるパーツ・カタログでは膨大な数量となる。そしてこれは全産業にも通じることである。

これらを立体図形でコンピュータへライブラリとして登録しておけば，大きさ，角度はもちろん，数量の制限もなく，統一性の高い部品を組立て工場の部品のように利用できる。

簡単で単純な仕事を解除できるのがコンピュータである。部品のデータベース化は限りなく前進させることが可能であり，

将来の多目的利用，コスト低減にも期待がもてるのである(**図2**)。

■立体図のデータベース化

パーツのデータベース化は分解図を作成するための補助ツールであり，真のデータベース化の効果は作成された分解図に潜んでいる。あらゆる機械装置類は，常に改善・更新を繰り返す。立体図にもこのサイクルがあり，デジタル化された立体図情報はこの段階で初めてコンピュータ時代に沿った利用形態を歩み出すことが可能になるのである(**図3**)。

■データベース化される立体図の可能性

前述したように，現在，立体図面の大半の利用はサービス部門においてである。サービス部門における目的と効果は先刻承知の通りである。

立体図作成の時期は，開発設計との並行作業も可能であるし，試作設計完了時でも同様である。メーカーの開発部門にCAD化が進行すると，設計データを即，磁気テープ，磁気ディスク，あるいは通信で入手できる。サービス部門で発揮されている効果は，製造企業の生産技術，部品の手配，資材手配，作業要領書などの分野においても有効な役割が果たせる。

コンピュータ化によって立体図作成が簡単にできる方向性を示し始めている。今，立体図の作成を思い切り企業活動の上流に位置付けし，媒体としての最大の利点を生かす運動と活動が必要になっている。四半世紀に及ぶ手描き作成時代に果たせなかった立体図の多目的利用が，コンピュータによって今，実現しようとしている。

■CAD化された立体図とDTP時代

コンピュータ技術の発展は目覚ましく，16ビットから32ビット，デスクトップからラップトップへと目的・用途に合わせて選択できる時代に突入した。特に，ドキュメント・マニュアル類の文書を作るためのDTPの普及は，タイプ，写真植字機，ワードプロセッサの領域を急速に凌駕し始めている。種々のドキュメント類には必ず説明図が挿入される。特にテクニカル・ドキュメント類は，説明図の内容によって理解力が左右される。手描き図をイメージ・スキャナで入力し，編集を行うことは工程の多さ，データの要素量において致命的欠点をもっている。CADによって作成されたベクトルデータは同じコンピュータのハードディスクの中でファイルでき，交互にデータのコンバートが可能になり，DTP時代の編集効果，工程に大きな変革をもたらすことが可能である。手描き技術，スキャナ，CADを巧みに組み合わせれば，熟練技術者しかできなかった写真トレース業務が短時間に，新人でもベクトル化することも可能になっている(**図4**)。

■まとめ

国内における立体図の本格利用から25年，今，軸測投影法がコンピュータ化のために体系化された画法であったと実感として受け止めている。

手で描く時代は終わった。これは改めて認識することであり，否定しても仕方のないことだと思う。

今後の四半世紀を生きていくための生活があり，顧客の要望にこたえ，生産の継続責任もある。

今日からでもコンピュータやCADソフトについて真剣に研究し，新しい生産体系を作り，立体図がコンピュータ化時代の情報伝達媒体として利用される環境を創造し，確かな自信と，立体図作成の貴重なスキルを生かして産業界に貢献していきたいものである。

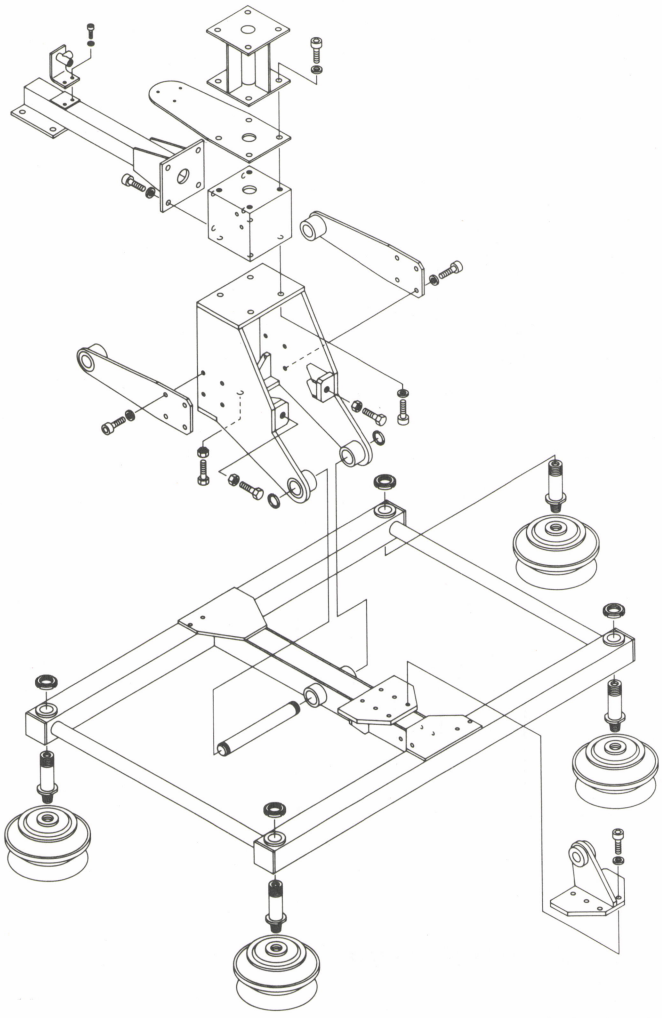


図3　立体分解図の作成例

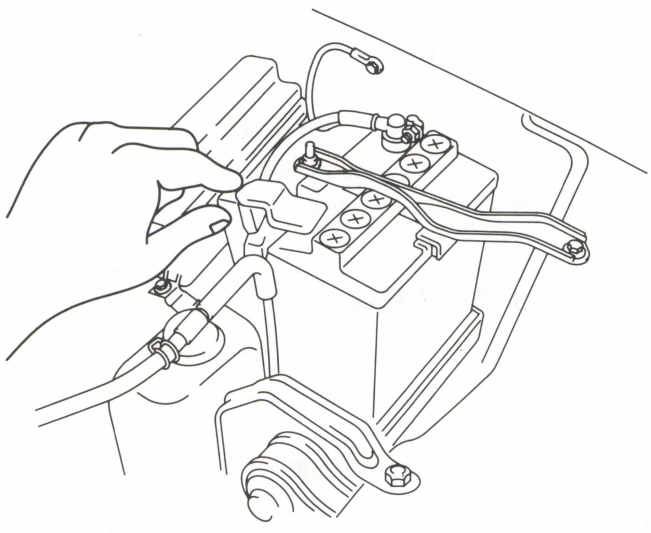


図4　CADによる写真トレース作成例

連載

CGのための図学(10)

コンピュータ・グラフィックスで扱われる図学――図形処理の基礎学について，プログラムリストを示しながら説明する。

長島　忍*

はじめに

これまで自由曲面を取り上げたときに，等角投影法による表示を行ったが，表示の際の投影法や等角投影法についてはまだ説明していなかった。何回か自由曲線・曲面の説明が続いたので，今回から表示法や投影法について説明する。表示というと，稜線表示，面表示，隠線消去表示，レンダリング表示などいろいろな表示があり，形状モデルの表現方法も含めなければならないので，ここでは形状はワイヤー・フレームなどの簡単な構造を対象にしておく。今回は，3面図や等角投影の表示方法について説明する。

投影法の種類

3次元の形状に関する情報を表示装置に出力するために，3次元の座標を2次元に変換するのが投影である。よく知られているように，投影には中心投影と平

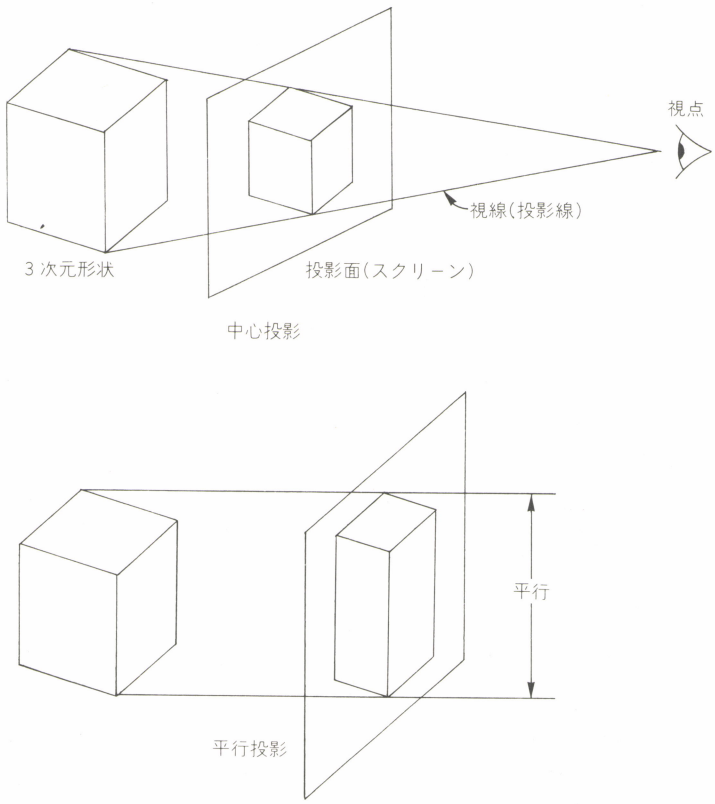
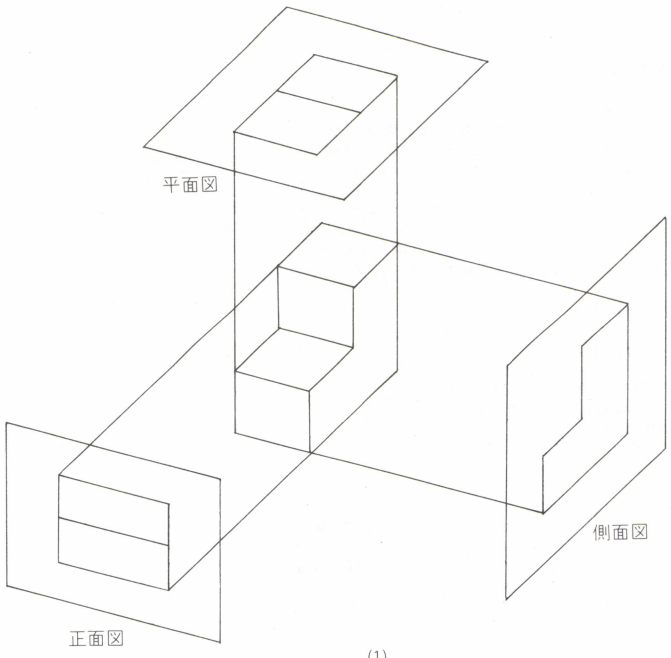
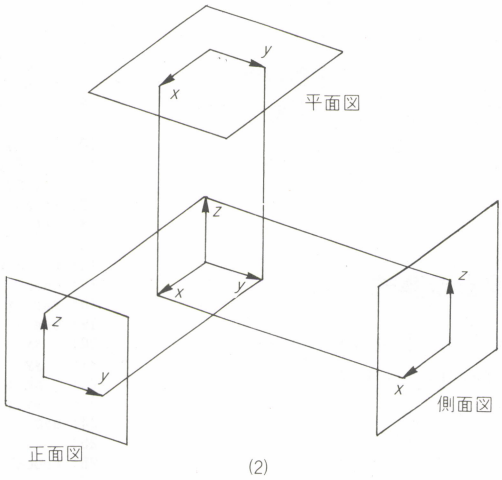


図1　中心投影と平行投影

* ながしま　しのぶ　東京大学　教養学部　情報・図形科学教室　〒153　東京都目黒区駒場3-8-1





◀図 2　3 面図

行投影がある(**図 1**)。普通の表示では，人が物を見た感じに近い中心投影－透視図が用いられるが，平行投影も図面としての表示に用いられる。

図面で代表的なのが 3 面図であり，3 面図は**図 2** (1)のように，正面図・平面図・側面図の 3 通りの投影を組み合わせた表示である。このような投影を正投影または主投影という。正投影は図 2 (2)のように座標値の 2 成分を選び，拡大および平行移動を行って画面の座標値として表示すればよい。

立体の回転を行えば任意方向からの表示ができるが，斜めの方向から見た投影を行う軸測投影図の例を**図 3** に示す。コンピュータを使えば任意方向からの投影は容易に計算・表示できるが，手書きの製図では作図が面倒なので，従来から簡単に見取図などを作図できる軸測図がよく用いられた。図 3 は 1 辺の長さが 1 の立方体を表示した場合の投影面における辺の長さと角度を示したもので，三軸測図，二軸測図，等測図，カバリエ投影図，ミリタリ投影図などがある。等測図は特によく用いられ，等角投影図とよばれている。

なお三軸測図，二軸測図，等測図は直

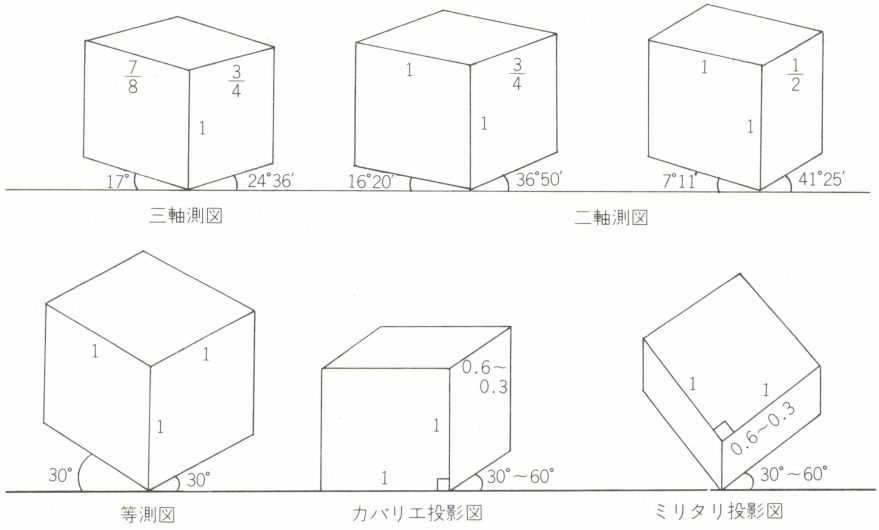


図 3　軸測図の例

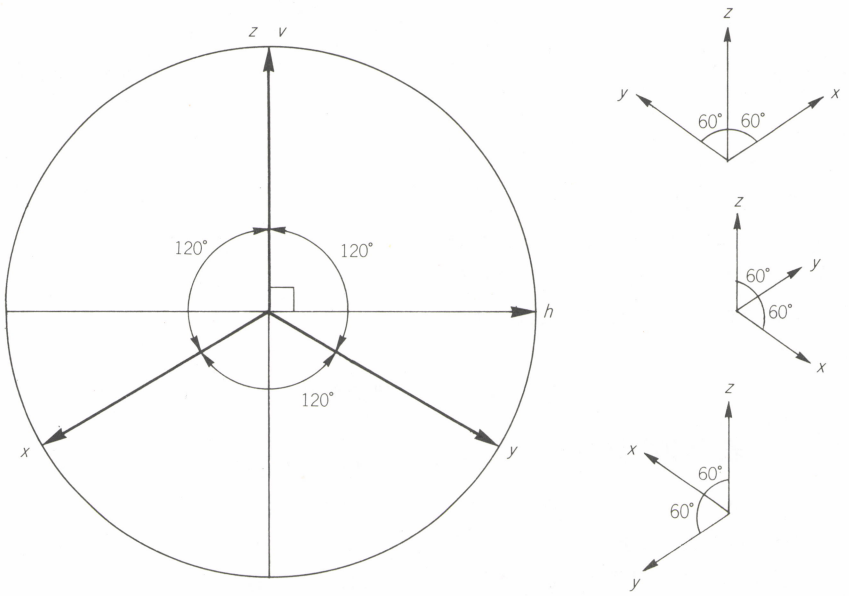


図 4　等角投影法(等測図)

◉図4は前ページ。

軸測図であり，投影線(視線)と投影面が垂直に交わっている。カバリエ投影図，ミリタリ投影図は垂直に交わっていないので，斜軸測図といわれる。正投影図や直軸測図は直投影，斜軸測図は斜投影とよばれ，直投影の変換は直交行列で表される。

投影変換式

ここでは，正面図・平面図・側面図の3面図の表示を行う投影法と等角投影法の変換式について説明する。

xyz 軸が図2(2)のように表示されると考え，スクリーン上の座標系を(h, v)とすると，正面図では，

$h=y \quad v=z$

平面図では，

$h=y \quad v=-x$

側面図では，

$h=-x \quad v=z$

になる。(h, v)の座標系はさらに，ディスプレイの座標系に合わせるための拡大・縮小と平行移動が必要になる。

等角投影図は xyz 軸が**図4**のように表示される変換であり，スクリーン上の座標系を(h, v)とし，図において各軸の長さを1とすると，x 成分だけに注目すれば，

$h=-\sqrt{3}/2x \quad v=-1/2x$

同様に y 成分については，

$h=\sqrt{3}/2y \quad v=-1/2y$

z成分については，

$h=0 \quad v=z$

となる。したがって，任意の(x, y, z)についてはそれぞれの和をとればよいから，

$h=-\sqrt{3}/2x+\sqrt{3}/2y$

$v=-1/2x-1/2y+z$

となる。3面図同様，ディスプレイの座標系に合わせるための拡大・縮小と平行移動が必要である。等角投影図には右側の3種類の投影法もある。

表示プログラム

ワイヤー・フレームで表された立体を3面図と等角投影図で表示するプログラムを，**図5**に示す。対象とする立体は，稜線にあたる線分のデータだけをもったワイヤー・フレーム・モデルとよばれる構造を用いる。実際の応用分野では，面のデータをもったサーフィス・モデルや中実の立体を表現したソリッド・モデルなどが利用される。

ワイヤー・フレームの立体を表現するには，稜線の数とそれらの線分の両端点の座標値のデータを与える。立体のワイヤーのデータは始点・終点の座標値の組の集合である。プログラムの中にデータを入れたが(7〜14行)，ファイルから読み込むのが普通である。キーボードから入力するのは，キーの打ち間違えや再実行を考えるとかなり面倒である。

メインプログラム(38〜63行)では，稜線に相当するワイヤーのデータを4種類

```
 1: {  *** ワイヤフレームの三面圖・等角投影圖表示 ***  }
 2: program isometric;
 3:
 4: {$I plot.lib}       {  グラフィクス・ライブラリ  }
 5:
 6: type dim = array[1..3] of integer;
 7: const now = 15;      {  ワイヤ数と端点の座標値  }
 8:       wire : array[1..now,1..2,1..3] of integer = (
 9:  ((0,0,0),(100,0,0)),((100,0,0),(100,150,0)),((100,150,0),(0,150,0)),
10:  ((0,150,0),(0,0,0)),((0,0,0),(0,0,150)),((0,0,150),(50,0,150)),
11:  ((50,0,150),(50,0,50)),((50,0,50),(100,0,50)),((100,0,50),(100,0,0)),
12:  ((100,0,50),(100,100,50)),((100,100,50),(50,100,50)),
13:  ((50,100,50),(50,0,50)),((0,0,150),(0,150,0)),
14:  ((50,0,150),(50,100,50)),((100,100,50),(100,150,0)));
15:
16: procedure isome(vtx : dim; var hh,vv : integer);  {  等角投影圖  }
17: const scl = 1.0; sq32 = 0.8660254;
18: var  xx, yy : real;
19: begin
20:   xx := -sq32*vtx[1] + sq32*vtx[2];
21:   yy := -0.5*vtx[1] - 0.5*vtx[2] + vtx[3];
22:   hh := round( xx*scl + 500.0);
23:   vv := round(-yy*scl + 200.0);
24: end;
25:
26: procedure orth(vtx : dim; var k,hh,vv : integer);   {  正投影  }
27: const cx : dim = ( 50, 50,350);     {  中心座標  }
28:       cy : dim = (100,380,380);
29:       xyz1 : dim = ( 2, 2, 1);      {  xyz成分の選択  }
30:       xyz2 : dim = ( 1, 3, 3);
31:       sclh : dim = ( 1, 1,-1);      {  xyz軸の向き  }
32:       sclv : dim = ( 1,-1,-1);
33: begin
34:   hh := sclh[k]*vtx[xyz1[k]] + cx[k];
35:   vv := sclv[k]*vtx[xyz2[k]] + cy[k];
36: end;
37:
38: var sp, ep : dim;              {  メインプログラム  }
39:     i, j, k, h1, v1, h2, v2 : integer;
40: begin
41:   g_init;
42:   g_cls;
43:   for i := 1 to now do begin    {  線分描画の繰り返し  }
44:     for j := 1 to 3 do begin
45:       sp[j] := wire[i,1,j];
46:       ep[j] := wire[i,2,j];
47:     end;
48:     for k := 1 to 4 do begin
49:       if k < 4 then begin
50:         g_color(4);             {  k=1  正面圖  }
51:         orth(sp,k,h1,v1);       {  k=2  平面圖  }
52:         orth(ep,k,h2,v2);       {  k=3  側面圖  }
53:       end
54:       else begin
55:         g_color(7);
56:         isome(sp,h1,v1);        {  k=4  等測圖  }
57:         isome(ep,h2,v2);
58:       end;
59:       g_line(h1,v1,h2,v2);
60:     end;
61:     read;          {  リターンキーを押す  }
62:   end;
63: end.
```

図5　3面図・等角投影図表示プログラム

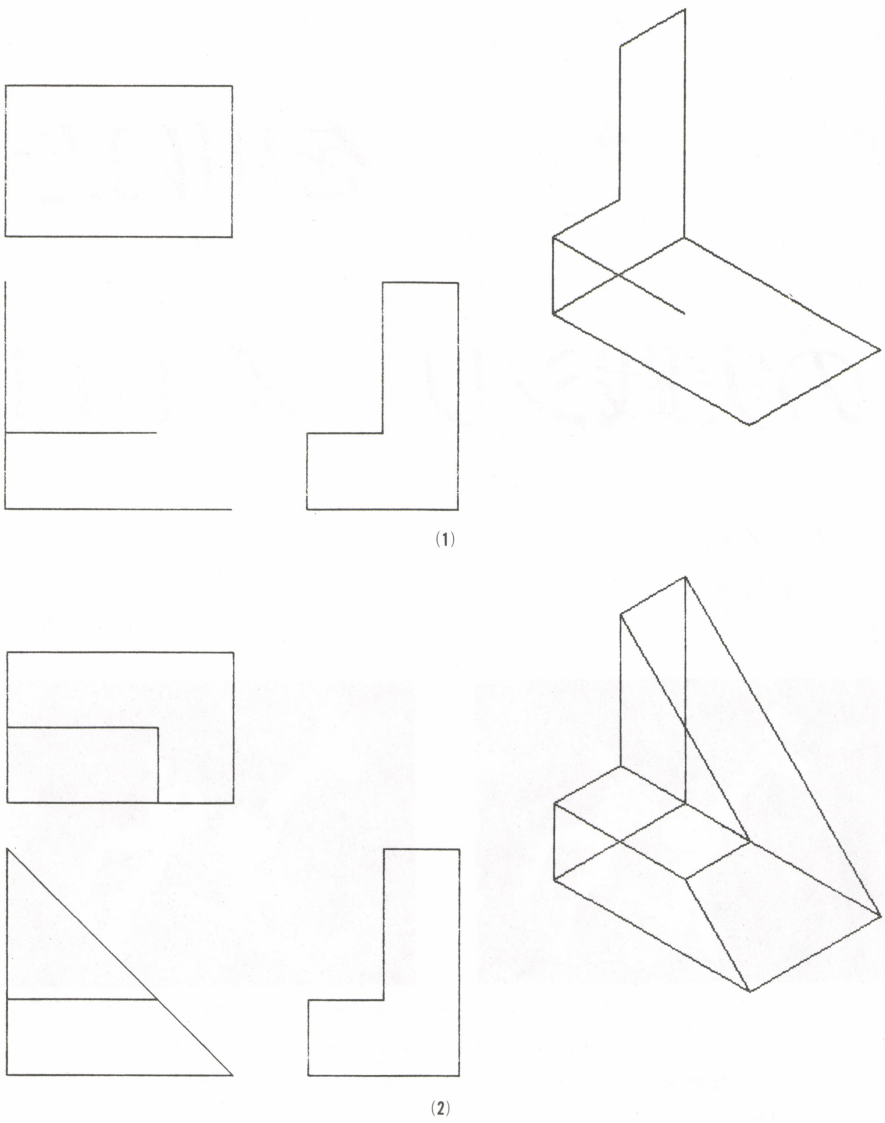


図6　プログラムの表示結果

の変換を行ってそれぞれ表示する。配列sp，epは3次元空間における始点・終点の座標値である(45～36行)。変数kの値により，正面図・平面図・側面図・等測図のための投影変換を手続きに行わせる。h1，v1，h2，v2はディスプレイ座標系における始点・終点の座標値である。稜線1本ごとにリターンキーを入力するようになっているので(61行)，1つ1つの稜線が各投影においてどのように表示されるかがわかる。

手続きisome(16～24行)，orth(26～36行)は，それぞれ等角投影図，3面図を表示するための手続きで，3次元の実数型の座標値を画面上の2次元の整数型の座標値に変換する。定数sq32は$\sqrt{3}/2$の値である。正面図，平面図，側面図，等角投影図の中心座標は画面上でそれぞれ(50, 100)，(50, 380)，(350, 380)，(500, 200)になっている。

なお，コメント{　}の中の'図'という文字の代わりに古い漢字が使われているが，これは'図'のJISコード(シフトJISコード)の第2バイトが'}'と同じであるためである。コメントの記号を{　}から(*……*)に変えればよい。

このプログラムの出力結果を**図6**に示す。図6(1)は途中の表示で，図6(2)が最終的な出力図である。

練習問題の解答

前回(88年12月号)の練習問題[9-1]の答えについて説明する。これ以外については，次回以降説明することにする。

● [9-1]　3次Bézier曲線$\boldsymbol{P}_1=(0,0)$，$\boldsymbol{P}_2=(3,0)$，$\boldsymbol{P}_3=(0,3)$，$\boldsymbol{P}_4=(3,3)$を$t=1/3$，2/3で3つの曲線に分割すると?

3次Bézier曲線$\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4$をパラメータ$t$で$\boldsymbol{a}_1\boldsymbol{a}_2\boldsymbol{a}_3\boldsymbol{a}_4$と$\boldsymbol{q}_1\boldsymbol{q}_2\boldsymbol{q}_3\boldsymbol{q}_4$に分けると，

$$\boldsymbol{a}_1=\boldsymbol{P}_1$$
$$\boldsymbol{a}_2=(1-t)\boldsymbol{P}_1+t\boldsymbol{P}_2$$
$$\boldsymbol{a}_3=(1-t)^2\boldsymbol{P}_1+2(1-t)t\boldsymbol{P}_2+t^2\boldsymbol{P}_3$$
$$\boldsymbol{a}_4=(1-t)^3\boldsymbol{P}_1+3(1-t)^2t\boldsymbol{P}_2+3(1-t)t^2\boldsymbol{P}_3+t^3\boldsymbol{P}_4$$
$$\boldsymbol{q}_1=\boldsymbol{a}_4$$
$$\boldsymbol{q}_2=(1-t)^2\boldsymbol{P}_2+2(1-t)t\boldsymbol{P}_3+t^2\boldsymbol{P}_4$$
$$\boldsymbol{q}_3=(1-t)\boldsymbol{P}_3+t\boldsymbol{P}_4$$
$$\boldsymbol{q}_4=\boldsymbol{P}_4$$

であるから，上記の曲線をまず$t=1/3$で2つの曲線に分割すると，

$\boldsymbol{a}_1=(0,0)$，$\boldsymbol{a}_2=(0,1)$，
$\boldsymbol{a}_3=(4/3,1/3)$，$\boldsymbol{a}_4=(13/9,7/9)$
$\boldsymbol{q}_1=(13/9,7/9)$，$\boldsymbol{q}_2=(5/3,5/3)$，
$\boldsymbol{q}_3=(1,3)$，$\boldsymbol{q}_4=(3,3)$

になる。次に曲線$\boldsymbol{q}_1\boldsymbol{q}_2\boldsymbol{q}_3\boldsymbol{q}_4$をさらに2つに分割するのでパラメータ値1/2で$\boldsymbol{b}_1\boldsymbol{b}_2\boldsymbol{b}_3\boldsymbol{b}_4$と$\boldsymbol{c}_1\boldsymbol{c}_2\boldsymbol{c}_3\boldsymbol{c}_4$に分割すると，

$\boldsymbol{b}_1=(13/9,7/9)$，$\boldsymbol{b}_2=(14/9,11/9)$，
$\boldsymbol{b}_3=(13/9,16/9)$，$\boldsymbol{b}_4=(14/9,20/9)$
$\boldsymbol{c}_1=(14/9,20/9)$，$\boldsymbol{c}_2=(5/3,8/3)$，
$\boldsymbol{c}_3=(2,3)$，$\boldsymbol{c}_4=(3,3)$

となる。したがって$\boldsymbol{a}_1\boldsymbol{a}_2\boldsymbol{a}_3\boldsymbol{a}_4$，$\boldsymbol{b}_1\boldsymbol{b}_2\boldsymbol{b}_3\boldsymbol{b}_4$，$\boldsymbol{c}_1\boldsymbol{c}_2\boldsymbol{c}_3\boldsymbol{c}_4$で表される3つの曲線が，$\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4$を3分割した曲線群である。ちなみに，もとの曲線$\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4$を$t=2/3$で分割した後半の曲線$\boldsymbol{d}_1\boldsymbol{d}_2\boldsymbol{d}_3\boldsymbol{d}_4$は，

$\boldsymbol{d}_1=(14/9,20/9)$，$\boldsymbol{d}_2=(5/3,8/3)$，
$\boldsymbol{d}_3=(2,3)$，$\boldsymbol{d}_4=(3,3)$

となって$\boldsymbol{c}_1\boldsymbol{c}_2\boldsymbol{c}_3\boldsymbol{c}_4$に完全に一致することがわかる。

練習問題

[10-1]　任意方向からの平行投影の変換式は?

[10-2]　等角投影図の軸の縮小率は?

[10-3]　等角投影図の残りの3つ(図4)の変換式は?

[10-4]　正投影，斜投影の変換行列は?

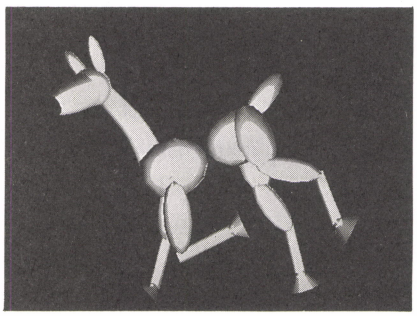
[10-5]　頂点に番号をつけたワイヤー・フレームのデータ構造は?

連載

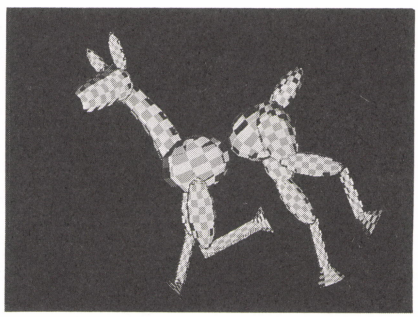
Cプログラミングを用いたレンダリング・ソフトの実践シリーズ〔4〕

今回のシリーズの最終段階として，スムーズ・シェーディングとマッピングまで可能なデプス・バッファのソース・プログラムについて解説する。

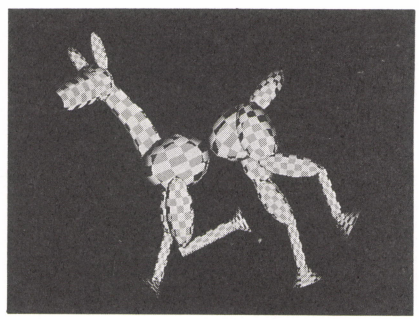
出渕　亮一朗*



① スムーズ・シェーディングのみ



② マッピングのみ



③ スムーズ・シェーディング＋マッピング

debut ver.2 の概要

またしても前回（88年11月号）の予告と違うのだが，今回はデプス・バッファによるレンダリング・アルゴリズムを解説するこのシリーズの最終段階として，debut ver.1 を発展させ，ver.2 として Phong のスムーズ・シェーディング，そしてテクスチャ・マッピングまで可能なバージョンのサンプル・プログラムとその解説を行いたいと思う。

ソース・プログラムの使用方法は前回の ver.1 と同様である。すなわち，

- ver.1 のヘッダ surf.h の代わりに，今回はそれをバージョンアップした phong.h を使用する。
- main ( ) を今回，再定義する ver.2 用 main（ ）に置き換える。
- 今回新たに定義する関数群を ver.1 で使用した関数群にすべて付け加える。

ただし，例えば ReadObjFile2（ ）のように，2が関数名の最後に付けられている関数は，それが付けられていない関数（この場合，ReadObjFile（））の一部を修正したバージョンアップ版であり，こうした関数についてはもとの関数はプログラム内で実際は使用していない。もしも，余計な関数でソースの長くなるのが気になる人は，それらを取り除いてもらってもかまわない。

debut ver.2 のアルゴリズムの焦点は頂点属性の中割りである。今回のシリーズのようなデプス・バッファやスキャンライン・アルゴリズムでは，通常，ポリゴンをベースとしている。ポリゴンはいくつかの頂点の連なりで表されるのだが，各頂点にはさまざまな属性を割り当てることができる。頂点座標 (x, y, z, w) は当然であるが，その他，頂点法線ベクトル（x, y, z），頂点カラー値（r, g, b, m），頂点マッピング座標（u, v, w）なども割り当てることができる。属性とは，オブジェクトの表面特性を与えるすべてのパラメータについて考えることができる。頂点をサンプル点として，それにそれら属性を適当にもたせておき，それ以外のポリゴン上の点ではサンプル点の属性をリニア補間して疑似的に属性を与えようという考え方である。このように，スムーズ・シェーディングもマッピングも考え方は基本的に同じである。

頂点法線ベクトルを補間することによりスムーズ・シェーディングを行うのが Phong の法線ベクトル補間濃淡付け（normal vector interpolation shading）であり[1]，頂点明度を補間することによりスムーズ・シェーディングを行うのが Gouraud の明度補間濃淡付け（intensity interpolation shading）である[2]。また，マッピング座標を補間して各ピクセルにそのマッピング座標に対応し

*でぶち　りょういちろう　〒154　東京都世田谷区三宿 1-15-11　コート世田谷 101

◉図2は次ページ。

た明度値をマッピング・テーブルから拾ってくるのが，ポリゴンをベースとした場合のいわゆるテクスチャ・マッピングである。Phongのスムーズ・シェーディングはGouraudによるものよりもマッハバンド問題を減じているといわれる。そのため，debut ver.2ではPhongによるものを採用し，色を用いた補間は行っていない。ただし，光によるスムーズ・シェーディングとは別にオブジェクトのカラーリングの特殊な表現として，各頂点に好きな色を割り当て，それを補間するという手法にGouraudは使用できると思う。(Northern Arizona UniversityのCraig Caldwellの作品，"Looking In"でのにじんだ虹のような色彩をしたロッキングチェアに腰掛ける人物（サイボーグ?）のような効果をつけることができる。)

連載第1回（88年6月号）の中で，debutを用いてスムーズ・シェーディングを行う場合，すべてのポリゴンを三角分割する必要があると仕様を決めたが，実際は三角形以上の多角形でも行えるようにプログラムは書かれている。ただし，三角形でない場合は凹多角形の場合などきれいにスムーズ・シェーディングがかからない場合があるので，やはり三角分割されていることが好ましい。

debut ver.2で書き直された関数のほとんどは，それまでの関数にこの頂点属性を付け加える必要があったものが大部分である。属性をもった頂点については，今回のphong.hの中でPointというtypedefに定義してある。CalPolygonEdge2（ ），CalPolygonRegion2（ ），CalPolygonCenter2（ ），CalPolygonNormal2（ ），CalPixelColor（ ）などが，Pointを付け加えられた関数である。

PolygonClip2（ ），CalCrossWall2（ ）は，頂点属性をもつポリゴンがクリッピング・ウィンドでクリッピングされた場合，再定義されるクリッピングされたポリゴンの頂点を考えれば，そのウィンド枠上にある頂点はもとのウィンド内外にあった頂点属性の補間された属性をもっていなければならないはずである。そのため，以前の関数を手直しして新たにクリッピングのための関数を定義している。

SmoothPolygonProcess（ ），PolygonScanConvert2（ ）は，スムーズ・シェーディングとマッピングのためのメインルーチンである。InBetweenMidAtribute（ ）は，ある線分の両端のPointを中割るPointをそれぞれのy座標値をもとにして求めるものである。InBetweenPixelAtribute（ ）は，同様のことをx座標値をもとにして実行するものである。OpenMapFile（ ），CloseMapFile（ ），CalMapFile（ ）はマッピングのための関数群である。

頂点属性の補間

図1のようにポリゴン$ABCD$があるとする。そしてスクリーン座標yをとるスキャンラインが図のように線分ADの分割点Iから進入し，線分BCの分割点Rから退出するものとする。各頂点の属性を例えばA_t，スクリーン座標をA_x，A_yのように表すとすると，

$$I_t=D_t+(A_t-D_t)(I_y-D_y)/(A_y-D_y)$$
$$R_t=C_t+(B_t-C_t)(R_y-C_y)/(B_y-C_y)$$
$$P_t=I_t+(R_t-I_t)(P_x-I_x)/(R_x-I_x) \quad (1)$$

である[3]。

まず線分ADのスキャンラインによる分割点Iの属性を頂点A，Dとスキャンラインのy座標から補間して求め，線分BCから同様に分割点Rの属性を補間して求める。最後に，線分IRの分割点Pの属性を点I，R，Pのx座標により補間して求めているのである。こうして(1)式により，スキャンライン上の各ピクセルPの属性P_tがリニア補間によって求められる。

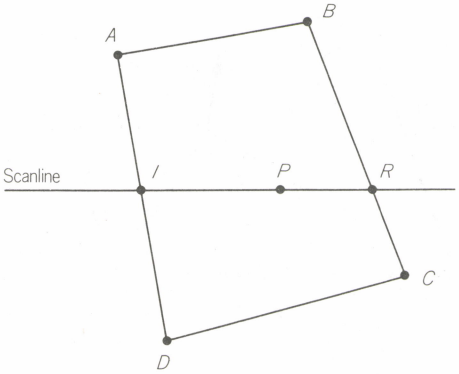


図1　頂点属性の補間

レンダリング・アルゴリズムにおいては，視点座標系に透視変換を行い，正規透視座標系に直してからポリゴンのスキャン・コンバートを行うのが普通である。それは，前回説明したようにポリゴンのデプスソートが楽であり，また頂点属性の中割りを考える場合においても有意義である。仮に，視点座標系で上記の式によりリニア補間をすることを考えると，次のような面倒が生じる。

もしも，ポリゴン$ABCD$が投影スクリーンに平行である場合はどうなるか？例えば，線分ADがスクリーンx座標に垂直であるとすれば，ポリゴン$ABCD$は可視であるにもかかわらず線分ADと線分BCは同一x座標上に重なり合ってしまい，x座標をもとにしたリニア補間を行うことができない。しかし，透視変換を行っていればこの問題は生じない。もし，ポリゴン$ABCD$が投影スクリーンに垂直であれば，そのポリゴンの法線ベクトルは視線と垂直であり，ちょうど見えない場合であるから，back facing polygon removal（前回説明した）の機能により，そのポリゴンは消去されているはずであるからである。

しかし，透視変換を行っていると，また新たな問題が生じる。それは透視変換により空間のひずみが生じ，正しいリニア補間が(1)式によって得られないということである。A_e，B_eを視点座標系内の2点とし，その透視変換後の対応点をA_p，B_p，スクリーンへの投影をA_s，B_sとする。スクリーン上で線分A_sB_sのある分割点をM_sとする。透視座標内ではx，y値については，$A_s=A_p$，$B_s=B_p$，$M_s=M_p$である。ところが，$S_p=|M_p-A_p|/|B_p-A_p|$とすると，**図2**のように透視変換前のS_eを考えれば，$S_e=|M_e-A_e|/|B_e-A_e|$でA_eとB_eが同一Z値でなければ$S_e\neq S_p$となってしまう。これでは，A_s，B_sを補間に利用するかぎりA_eとB_eの間をうまくリニア補間できないことになる。これを解決するためには，A_p，B_pの同次座標のw値を使ってもとのA_e，B_eに戻して考えなければならない。だが，M_pの対応点M_eは未知である（もし視点座標系内なら，これは光線追跡で求められるであろ

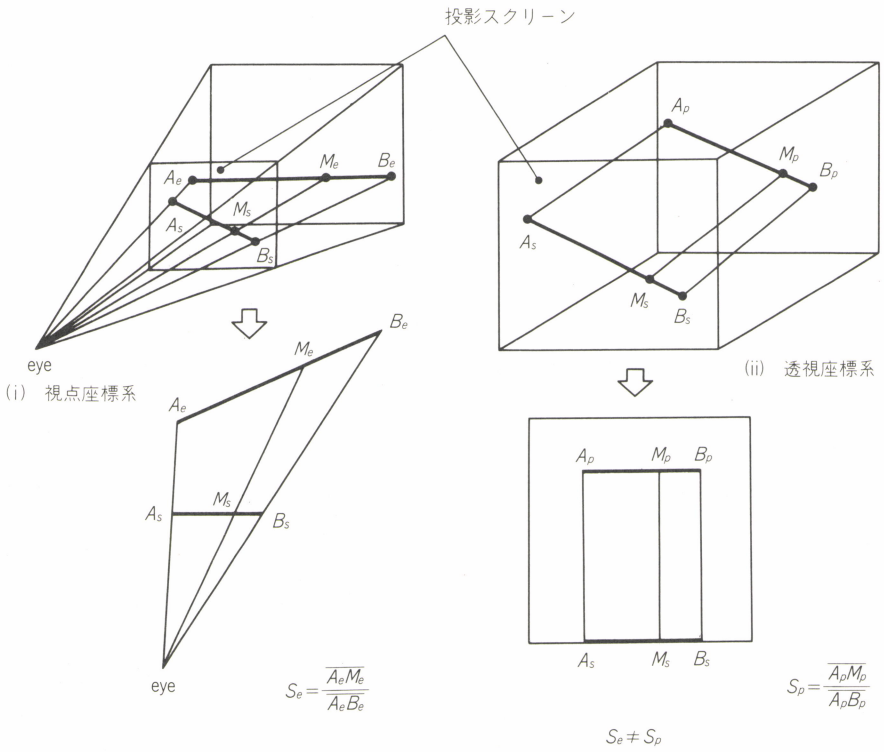


図2 透視変換によるひずみ

うが)。しかし，これらは M_eが A_eと B_eをどのような比で分割しているかという値 S_eがわかれば求めることができるはずである。実際 S_eは結果だけ示せば次の式で表される。

$$S_e=A_w \cdot S_p/((B_w-A_w)\cdot S_p-B_w) \quad (2)$$

ただし，

$$[x_e\ y_e\ z_e\ 1]T_p=[X_p\ Y_p\ Z_p\ w_p]$$
$$x_p=X_p/w_p$$
$$y_p=Y_p/w_p$$
$$z_p=Z_p/w_p \quad (3)$$

T_pは透視変換マトリクスとする[4]。

つまり，各A，B点のw値を用いてS_eが求められるということである。この S_eを用いて，

$$M_t=A_t+S_e\cdot(B_t-A_t) \quad (4)$$

と M 点の属性をニリア補間することができる。CorrectDistortPoint ( ) がこのひずみを修正するための関数である。

スムーズ・シェーディング

前述したように，頂点属性の補間によるスムーズ・シェーディングにはGouraudによるものとPhongによるものがある。Gouraudの明度補間濃淡付けは，各頂点における明度を前回示した表面色決定法と同様の方法で行い，その後，それぞれの頂点間の明度を第2項で示した補間法により決定するものである。しかし，Gouraudの方法ではマッハバンドの問題が生じる場合がある。Phongの法線ベクトル補間濃淡付けは，各頂点の明度の代わりに頂点における法線ベクトルを利用するものである。前回示したように，明度決定にかかわるdiffuse値もspeculer値も物体表面の法線ベクトルにかかわっている。レイ・トレーシング法においても追跡中の光線とプリミティブとの交点における法線を求め，それを用いてその点での明度を計算している。この法線ベクトルを各頂点でリニア補間することにより，すべての表面上の明度を滑らかに決定できるのである。

頂点法線ベクトルを求めるためには次の手順を踏めばよい（**図3**）。

- 現在の頂点 V に接続しているポリゴンをすべて求める。
- それらのポリゴンの法線ベクトルを求める(あらかじめ求めておいてもよい)。
- それらのポリゴンの法線ベクトルの平均を現在の頂点の頂点法線ベクトルとする。

すなわち，

$$\boldsymbol{N}_v=\sum_{i=0}^{i=n-1}\boldsymbol{N}_i/n \quad (5)$$

ただし，$\boldsymbol{N}_v$は頂点 V の頂点法線ベクトル。$\boldsymbol{N}_i$は V に接続する i 番目のポリゴン。n は V に接続するポリゴンの数。

PrePolygonProcess ( ) が，各ポリゴンの法線から頂点法線ベクトルを求めるためのものである。

頂点 A における法線$\boldsymbol{N}_a$と頂点 B における法線$\boldsymbol{N}_b$は通常リニアに補間するが，これは処理の高速化のためであり，本当にそれが一番よいかは別問題である。例えば，A と B に接し A と B での法線ベクトルを与えるある半径 R の円弧上の法線ベクトルを$\boldsymbol{N}_a$と$\boldsymbol{N}_b$を補間する法線ベクトルに採用した方が，あるいは滑らかなスムーズ・シェーディングが行えるのかもしれない。

また，この方法によるスムーズ・シェーディングの一番の欠点は，オブジェクトを構成するポリゴン数が十分多くない場合はいくら滑らかにシェーディングがかかっているようでも，そのエッジは多角形によるガタガタがそのまま残ってし

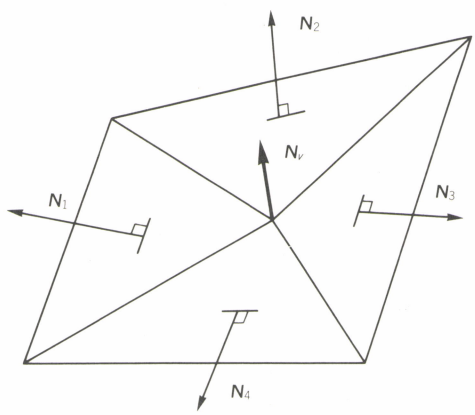


図3 頂点法線ベクトル

まうということである。これを解決するためにはポリゴン数を増やすしかないのだが，それはメモリの限界やCPUの問題などハードウエアの制限を大きく受けてしまう。これはポリゴンをベースとしたレンダリング・アルゴリズムの共通の問題点である。これを本質的に解決するためには，例えばPIXAR社が採用しているbicubic bézeir patchのような自由曲面によるパッチを用いるしかないであろう。(ちなみに，今年のSIGGRAPHでPIXAR社のRenderManの説明会でもその点を強調していたようだ。ところで，PIXAR machine自体にはモデリングやアニメーション・データを作成する機能はなく，本質的にレンダリングと画像処理のためのマシンである。しかしWavefrontのようにモデリングやアニメーション・データをワークステーション上でインタラクティブに作成できることは，最近のコンピュータ・グラフィックス(CG)ソフトウエアのスタンダードになりつつあると思う。RenderManのはじめの発想は，他のモデラーやアニメーション・ツールにより作られたレンダリング・データをPIXAR用のデータへとコンバージョンするソフトウエアを書くことを支援するツールであったような気がする。ついでのついでだが，Wavefrontはそのユーザーの作ったアプリケーション・ソフトを“Ripples”としてユーザー間で取引して提供し合うことを始めたようである。ハリウッドのCGプロダクション，ReZ-n8のメタモルフォーゼ・ソフト，Dv8などが主なところである。筆者のモデリング/アニメーション・ツール，Dynamation Systemもこれに載せることを計画中である。ただ，“Ripples”(さざ波)というネーミングが少々気に入らないが……)

マッピング

マッピングは，マトリクスとなったデータテーブル上のカラー値をオブジェクトの上の各点に対応させ，その点での明度を決定するものである。マッピング・テクスチャのためのデータテーブルは通常$x \times y$の縦横数をもった配列である。このx, yの最大，最小を0から1に対応させ，オブジェクトの上のuv座標値によってこのデータテーブルのある位置のデータを読み出してくればよいのである。各ポリゴンの各頂点になんらかの規則によってuv値を対応させるよう，あらかじめオブジェクト・データを作成しておけば，前述した方法でポリゴン上のすべての点におけるuv値をリニア補間により得ることができる(*uv*値の決定には極座標やオブジェクトを形作るメッシュへの対応を考えることが多い)。debutでは，第1回目でデータ形式定義した.objの中のMVがこのマッピング座標にあたる。ポリゴンデータSFの中で，(VX, MV)の形でポリゴンの各構成頂点の頂点座標とマッピング座標を結び付けている。

実際のプログラミング上の話であるが，このマッピング用のデータテーブルをプログラムの中に配列として持たせておくとすれば，莫大なメモリを必要としてしまう。概してマッピング・データは大きなものであり，またいくつものマッピングを使用する場合もよくあることである。この問題を解決するためにdebutではマッピング・データをディスク上にファイルの形として置いておき，プログラム中でファイルをオープンして必要に応じてlseek()でファイル中を検索し，必要なデータを拾ってくるよう工夫されている。これにより，ディスクの容量が許すかぎり大きなマッピング・データをいくつも用意できるのである。もちろん，ファイルの検索は処理時間上の問題がある。lseek()には，絶対アドレスでデータを読み出すモードと相対アドレスで読み出すモードとがある。絶対モードとは，例えばファイルの先頭から何バイト目か?という探し方をするのに対して，相対モードでは現在の位置からの相対的な移動量によりデータを読み出す。ポリゴンをスキャン・コンバートに従って各ピクセルの色を決定することを考えれば，アルゴリズム上，一つ前に探したマッピング・テーブルの位置と次に探すテーブルの位置には大きな相関があると考えてよい。これをもとにdebutではlseek()を相対モードで使用するよう工夫されている。明らかに絶対モードで使用するよりも高速化が図られていることが，実際にテストを行ってみて確かめられた。

debut ver.2で，一応今回のシリーズにおけるレンダリング・サンプル・ソフトの紹介は終わりとする。最後に，このソフトウエアを考えるうえでさまざまな助言を与えていただいた吉本圭一氏に感謝する(特に，視点座標系を透視座標系に変換することによるひずみに関する論議は彼の理論によるものである)。

次回は，最終回として前回予告したように，いくつかのカラー作品とそれを作るためのデータとその作成の仕方(特にツリー構造データの復習)，モデリングのための簡単なツールを紹介し，最後にCGツール環境を整えるための今後の発展に対するいくつかの助言をして，このシリーズを終わりにしたいと思う。

- debut ver.0,ver.1のちょっとしたバグをレポートする。
- ver.0のWirePolygonProcess()の中の/* put wire dump file */のコメントの後，fprintf(fp,"%f %f",*point〔j〕* view.sc2.x+view.sc2.x,*(point〔j〕+1) * view.sc2.y+view.sc2.y)；を，fprintf(fp,"%f %f",*point〔j〕* view.sc2.x+view.sc2.x,*(point〔j〕+1) * view.sc2.x+view.sc2.y)；に直す。
- ver.1のSurfacePolygonProcess()の中の/* viewport transformation */のコメントの後で，*(point〔j〕+1)=*(point〔j〕+1) * view.sc2.y+view.sc2.y;を，*(point〔j〕+1)=*(point〔j〕+1) * view.sc2.x+view.sc2.yに直す。

いずれも，ビューポートのy方向倍率をview.sc2.yからview.sc2.xに変更するものである。もしもこれらを行わなければ，縦方向の画面範囲以上に上下がクリッピングされた画像ができてしまう。

参考文献

1) Bui-Tuong Phong：“Illumination for Computer-Generated Pictures”, Communications of the ACM, Vol. 18, No. 6, pp. 311-317, 1975. 6
2) H. Gouraud：“Continuous Shading of Curved Surfaces”, IEEE Transactions on Computers, C-20, No. 6, pp. 623-628, 1971. 6
3) J. D. Foley, A. van Dam (今宮淳美訳)：「コンピュータ・グラフィックス」, 日本コンピュータ協会, 1984
4) 山口富士夫：「コンピュータ・ディスプレイによる図形処理工学」, 日刊工業新聞社, 1981

◉debut ver.0, ver.1, ver.2 は著作権法上，個人として利用する他は営利活動への使用，無断転載を禁じます。
COPY RIGHT©1988 R.DEBUCHI

```
/*
*       ************************
*       *    PHONG.H       *
*       ************************
*
*       header file for debut ver 2
*       (criate Phong-shading & mapping iamge program)
*/

#ifdef MAIN /* must define in main file*/

#define STRUCT struct
#define FLOAT float
#define INT int
#define SHORT short int
#define CHAR char

#else

#define STRUCT extern struct
#define FLOAT extern float
#define INT extern int
#define SHORT extern short int
#define CHAR extern char

#endif

#define MAXLINE         240  /* string buffer length */
#define MAXTREE         32   /* number of branches */
#define MAXPOLI         5000  /* number of poligons */
#define MAXOUT          5    /* number of outline vertice */
#define MAXEDGE         10   /* MAXOUT * 2 (for cliping) */
#define MAXPOINT        10000 /* number of points */
#define MAXSCREENX      640 /* max screen size x */
#define MAXSCREENY      640  /* max screen size y */
#define MAXLIGHT        16   /* number of lights */
#define MAXMATFILE      4    /* number of material files */
#define MAXLABEL        16   /* number of material labels */
#define MAXMATERIAL     36   /* number of materials */
#define MAXCONNECT      8    /* number of connect polygons or vertice */
#define MAXMAPVERT      10000     /* number of mapping vertex */
#define MAXMAPFILE      8    /* number of mapping file .map */

#define PAI 3.141592

#define EQUAL           0
#define TRUE            1
#define FALSE           0
#define NIL             -1
#define BYTE            255
#define ERR             -1
#define SMOLE           .000001
#define ON              1
#define OFF             0
#define HEADER          5

#define ABS(x) (x >= 0.0 ? x : -1.0 * x)
#define MAX(x, y)       ((x >= y) ? x : y)
#define MIN(x, y)       ((x < y) ? x : y)
#define SGN(x)          ((x >= 0) ? ((x > 0) ? 1 : 0) : -1)

typedef unsigned char U_Char;
struct U_buffer{
                        U_Char   buf[MAXSCREENY][MAXSCREENX];
        };

struct F_buffer{
                        float       buf[MAXSCREENY][MAXSCREENX];
        };

STRUCT U_buffer         Red, Green, Blue; /* color buffer */
STRUCT F_buffer         Depth; /* depth buffer */


/************************************/
/*   TreeData                       */
/* for node unit of tree-structure */
/************************************/

typedef struct{
                        U_Char   prio;
                        float       rot[3];
                        float       tran[3];
                        float       scale;
                } World;

typedef struct{
                        float       from[3];
                        float       at[3];
                        float       vrot;
                } Look;


struct TreeData{
                        char        label1[MAXLINE];
                        char        sign;
                        char        label2[MAXLINE];
                        char        name[MAXLINE];
                        char        type;

                        int         next;
                        int         lower;
                        int         obj;

                        float       matrix[4][4];
                        float       rmatrix[4][4];
                        float       mat[4][4];

                        union{
                                World     wld;
                                Look      look;
                        } p;
                };

STRUCT TreeData branch[MAXTREE];

/*************************/
/*   Object              */
/*   for object unit     */
/*************************/

struct Object{
                        char        name[MAXLINE];
                        int         tptr; /* pointer to branch */
                        int         pptr; /* pointer to vertex */
                        int         eptr; /* pointer to element */
                        int         mptr; /* pointer to material */
                        int         vptr; /* pointer to mapvert */

                        int         PointNum;
                        int         ElemNum;
                        int         MapVertNum;

                        char        usemtl[2][MAXLINE];
                        char        mtllib[MAXLINE];
};

STRUCT Object object[MAXTREE];


typedef struct{
                        float       x,
                                    y,
                                    z;
        } Vector3f;

typedef struct{
                float       r,
                            g,
                            b;
        } Color;

struct Light{
                        char            name[MAXLINE];
                        int             tptr; /* pointer to branch */
                        int             ltype; /* light type */
                        Color           color; /* illumiantion */
                        Vector3f        direct; /* direction */
                        Vector3f        posi;       /* position */
                        Vector3f        center; /* transformed position */
                        Vector3f        vec;        /* transformed direction */
                        float           rate; /* dumpping rate */
                        float           start; /* dumpping start point*/
};

STRUCT Light light[MAXLIGHT];

/*****************************/
/*   Vertex                  */
/* for vertex coordinates    */
/*****************************/


struct Vertex{
                        Vector3f        coord;
                        Vector3f        norm; /* normal vector */
                        short int       CntPolyNum;
```

```
                short int       cntpoly[MAXCONNECT]; /*connect polygon*/
        };

STRUCT Vertex vertex[MAXPOINT];

typedef struct{
                float   u,
                        v;
        } Vector2f;

struct  MapVertex{
                Vector2f        map;
        };

STRUCT MapVertex mapvert[MAXMAPVERT];

/*****************************************/
/*  Element                              */
/*  for outline connectivity of poligon  */
/*****************************************/

struct Element{
        short int       VertNum;
        short int       vert[MAXOUT];
        short int       mapv[MAXOUT];
        Vector3f        center;
        Vector3f        norm;
};


STRUCT Element element[MAXPOLI];

/*************************************************/
/*      Edge                                     */
/*      for one outline of individual polygon    */
/*************************************************/

typedef  struct{
                float   min;
                float   max;
        } Minmax;

typedef  struct{
                int     min;
                int     max;
        } Minmaxi;

typedef struct{
                Vector3f        posi; /* edge vertex coordinate */
                int             flag;
                Minmax          valy;  /* y-value min-max */
                int             vp0; /* pointer to vertex */
                int             vp1;
                float           cline; /* incriment x/y */
                float           zinc; /* incriment z/y */
        } Edge;
/***********************************************************
/*      Illum                                               */
/*      for relationship of lights and individual polygon */
/***********************************************************/

typedef struct{
                Vector3f        vec; /* vector:polygon -> light */
                float           distance; /* distance:polygon -> light*/
                float           dump; /* illumination dumped rate */
        } Illum;

/*****************************************
/*  Polygon                              */
/*      for     individual polygon      */
/*****************************************/

typedef struct{
                Vector3f        center;
                Vector3f        norm;
                Vector3f        center2; /* after perspective */
                Vector3f        norm2;  /* after perspective */
                float           cn;     /* center2 * norm2 */
                float           zinc;     /* incriment z/x */
                Color           color;
                Minmax          valy;     /* y-value min-max */
                int             EdgeNum;
                int             LitNum;
                int             mptr; /* pointer to material */
        } Polygon;

typedef struct{
                float   x,
                        y,
                        z,
                        w;
        } Vector4f;

typedef struct{
                Vector3f        coord; /* before perspective */
                Vector4f        coord2; /* after perspective */
                Vector3f        norm;
                Vector2f        map;
                Color           color;
        } Point;

/*********************************/
/*      Mapfile          */
/*    for mapping of objects       */
/*********************************/

typedef struct{
                int     x,
                        y;
        } Vector2i;

struct Mapfile{
                char            name[MAXLINE]; /* mapfile name*/
                Vector2i        size; /* mapfile size */
                FILE            *fp; /* FILE pointer */
                int             fd; /* FILE discripter */
                Vector2i        posi; /* current position */
                long            pos; /* file position */
                long            line; /* length 1 scan line */
                Color           color; /* current color */
        };
STRUCT Mapfile mapfile[MAXMAPFILE];

typedef struct{
                int             flag; /* mapping flag */
                char            name[MAXLINE]; /* mapping file */
                int             mptr;     /* pointer to mapfile */
                Vector2f        time; /* times */
                float           rate; /* mixing rate (color & map) */
        } Mapping;
/************************************/
/* Material                         */
/*      for material of objects     */
/************************************/

typedef struct{
                char    name[MAXLINE];
                int     mptr; /* pointer to material */
        } MatLabel;

struct  MaterialFile{
                int     MatNum;
                char    name[MAXLINE];
                MatLabel label[MAXLABEL];
        };

STRUCT MaterialFile matfile[MAXMATFILE];


struct Material{ /* not complete defined in this version */
                int     fptr; /* pointer to material file */
                int     lptr; /* order of new label */
                char    name[MAXLINE];
                int     lmodel; /* light model */
                Color   ambient;
                Color   diffuse;
                Color   speculer;
                Mapping ambmap;
                Mapping difmap;
                Mapping specmap;
                float   expo; /* exponent of speculer*/
                int     smooth; /* smoothing flag */
                int     map; /* mapping flag */
        };

STRUCT Material material[MAXMATERIAL];

/*********************************/
/*  View                         */
/*      for camera view          */
/*********************************/


typedef struct{
                float   left,
                        right,
                        bottom,
                        top;
        } Window;

struct View{
                char            name[MAXLINE];
                Vector3f        from,
                                at,
```

```
                        vector;
        float           heather;
        float           yon;
        float           angle;
        float           para1, /* heather / (screenx / 2) */
                        para2, /* heather / (screeny / 2) */
                        para3; /* yon / (yon - heather) */
        float           scale;

        Vector2i screen;
        Vector2i sc2;

        Window window;
};

STRUCT View view;

INT     ObjNum, TreeNum, LitNum, MatFileNum, MapFileNum;
INT     PointSum, ElemSum, MatSum, MapVertSum;
INT     const_fg, map_fg, smooth_fg;
INT     ambmap_fg, difmap_fg, specmap_fg;
FLOAT   pai;
FLOAT   ClipMat[4][4];
CHAR    ReadFile[MAXLINE], SaveFile[MAXLINE];



/*
*       **********************************
*       *          DEBUT  Ver 2 (Smooth)     *
*       **********************************
*
*       Written by: Ryoichiro Debuchi
*         (CATDOG - - is a brand name of all my works)
*       Date: Aug. 1988  c & p  R. Debuchi
*
*       Function:
*       Criate RGB color images.
*       By using of polygonal surface with smoothing & mapping.
*       Read structure file (.str) and put RGB file  (.rgb)
*       This version program consists of several main parts:
*       Read object file and tree-structure file,
*       modeling transformation by tree-structure,
*   view transformation, clipping, perspective transformation,
*   viewport transformation, scan-convert polygon,
*       calculate pixel color with lights ,smoothing, and mapping,
*       update depth and color buffers,
*       and output RGb color file.
*
*       Usage:
*               debut2 read.str write.rgb
*
*/

#include <stdio.h>
#include <math.h>
#define MAIN 1
#include "phong.h"

#define COMMENT         "#"
#define SWITCHD         "/D"
#define UPPER           "*"
#define ROOT            "ROOT"
#define DEFAULT         "default"

#define SAME            '|'
#define UNDER           '>'
#define CAMERA          'K'
#define WORLD           'O'
#define DAMMY           'D'
#define LIGHT           'L'

#define CONSTRUCT       "-O"
#define WIREDUMPCODE    "WireDump"

/*
*       ****************
*       *  main      *
*       ****************
*
*       Function:
*       main routine for debut ver 2.
*/

main(argc, argv)
int     argc;
char    *argv[];
{
        int     i;
        float   tm[4][4], rtm[4][4];
        FILE    *fp, *fopen();

        pai = PAI / 180.0;

        ClearLight();
        ClearMaterial2();
        ReadArg2(argc, argv);
        MatFileNum = 0;
        MapFileNum = 0;
        MapVertSum = 0;
        ReadTree(ReadFile);
        SetView();
        UnitMat44(ClipMat);
        ClipMat[0][0] = view.para1;
        ClipMat[1][1] = view.para2;
        RotEye(tm, rtm);
        Tree(branch, 0, tm, rtm);
        for(i = 0; i < LitNum; i++)
                WorldTransLight(light+i);
        ClearBuf();
        ClearVertex();
        for(i = 0; i < ObjNum; i++)
        {
                WorldTransVertex(object+i);
                PrePolygonProcess(object+i);
                SmoothPolygonProcess(object+i);
        }
        fp = fopen(SaveFile, "w");
        if(fp == (FILE *)NULL)
        {
                fprintf(stderr, "Can't Open File %s\n", SaveFile);
                exit(1);
        }
        OutputColorBuffer(fp);
        fclose(fp);
}



/* New Functions for Debut ver. 2*/

#include <stdio.h>
#include <math.h>
#include "phong.h"

#define VERTEX "VX"
#define SURFACE "SF"
#define MATERIAL "MT"
#define MAPVERT "MV"

/*
*       ***************
*       * ReadObjFile2 *
*       ***************
*
*       Function: read object file ver. 2 (with mapping vertice)
*       Usage:
*               struct Object *op (input) - pointer to object
*/

ReadObjFile2(op)
struct Object *op;
{
        char    tank[MAXLINE], line[MAXLINE];
        char    *p;
        int     i, n, m, q;
        int     vertn;
        int     en, pn, qn, mp;
        int     vn[MAXOUT], vtn[MAXOUT];
        float   vp[3];
        FILE *fp, *fopen();


        fp = fopen(op->name, "r");
        if(fp == (FILE *)NULL){
                fprintf(stderr, "Can't Open File %s\n", op->name);
                exit(1);
        }

        op->pptr = PointSum;
        op->eptr = ElemSum;
        op->vptr = MapVertSum;

        n = 0;
        m = 0;
        q = 0;

        while(1)
        {
                p = fgets(line, MAXLINE, fp);
                if(p == NULL)
                        break;
                sscanf(line, "%s", tank);
                if(strcmp(tank, VERTEX) == EQUAL){
                        pn = PointSum + n;
                        if(pn >= MAXPOINT){
```

```
                                fprintf(stderr,
                "Vertice too Large (<= %d) (At %s)\n", MAXPOINT, op->name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f %f", tank, vp, vp+1, vp+2);
                        (vertex+pn)->coord.x = vp[0] * 100;
                        (vertex+pn)->coord.y = vp[1] * 100;
                        (vertex+pn)->coord.z = vp[2] * 100;
                        n += 1;
                }
                else if(strcmp(tank, MAPVERT) == EQUAL){
                        qn = MapVertSum + q;
                        if(qn >= MAXMAPVERT){
                                fprintf(stderr,
                "MapVert too Large (<= %d) (At %s)\n", MAXMAPVERT, op->name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f", tank, vp, vp+1);
                        (mapvert+qn)->map.u = vp[0];
                        (mapvert+qn)->map.v = vp[1];
                        q += 1;
                }
                else if(strcmp(tank, SURFACE) == EQUAL){
                        en = ElemSum + m;
                        if(en >= MAXPOLI){
                                fprintf(stderr,
                "Face Outline too Large (<=%d) (At %s)\n", MAXPOINT, op->name);
                                exit(1);
                        }
                        vertn = GetElem(line, vn, vtn);
                        if(vertn > MAXOUT){
                                fprintf(stderr,
"Poingon Vertice too Large (<=%d) (%d SF line) (%s)\n", MAXOUT, m, op->name);
                                exit(1);
                        }
                        (element+en)->VertNum = vertn;
                        for(i = 0; i < vertn; i++)
                        {
                                *((element+en)->vert+i) = *(vn+i) + PointSum;
                                *((element+en)->mapv+i) = *(vtn+i) + MapVertSum;
                        }
                        m += 1;
                }
                else if(strcmp(tank, MATERIAL) == EQUAL)
                        MaterialFileProcess(line, op); /* add op 8/16 */
        }
        op->PointNum = n;
        op->ElemNum = m;
        op->MapVertNum = q;

        PointSum += n;
        ElemSum += m;
        MapVertSum += q;

        mp = op->mptr;
        if((material+mp)->map == ON){
                if(op->MapVertNum == 0){
                        fprintf(stderr,
                                "No Mapping Vertex !! (obj=%s)\n", op->name);
                        exit(1);
                }
        }
        fclose(fp);
}

#define NEWLABEL        "NEW"
#define LIGHTMODEL      "LTM"
#define AMBIENT         "AMB"
#define DIFFUSE         "DIF"
#define SPECULER        "SPC"
#define EXPONENT        "EXP"
#define SMOOTH          "SMS"
#define AMBMAP          "MAM"
#define DIFMAP          "MDI"
#define SPECMAP         "MSP"

/*
*       **********************
*       * ReadMaterialFile2 *
*       **********************
*
*       Function: reading material file ver. 2 (with mapping & smooth shading)
*       Usage:
*               char    *name   (input) - filename .mat
*/

ReadMaterialFile2(name)
char    *name;
{
        char    line[MAXLINE], tank[MAXLINE], lname[MAXLINE];
        char    tank2[MAXLINE];
        char    *p;
        float   val[3];
        int     lp, lm, l;
        int     ln = 0;
        int     l_fg = FALSE;
        FILE *fp, *fopen();

        fp = fopen(name, "r");
        if(fp == (FILE *)NULL){
                fprintf(stderr, "Can't Open File %s\n", name);
                exit(1);
        }
        while(1)
        {
                p = fgets(line, MAXLINE, fp);
                if(p == NULL)
                        break;
                sscanf(line, "%s", tank);
                if(strcmp(tank, NEWLABEL) == EQUAL){
                        ln++;
                        lp = ln + MatSum - 1;
                        sscanf(line, "%s %s", tank, lname);
                        strcpy((material+lp)->name, lname);
                        (material+lp)->fptr = MatFileNum;
                        lm = ln - 1;
                        (material+lp)->lptr = lm;
                        (matfile+MatFileNum)->label[lm].mptr = lp;
                        strcpy((matfile+MatFileNum)->label[lm].name, lname);
                        l_fg = TRUE;
                }
                else if(strcmp(tank, AMBIENT) == EQUAL){
                        if(l_fg == FALSE){
                                fprintf(stderr,
                                "New Material Label undefined (%s)\n", name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f %f", tank, val, val+1, val+2);
                        (material+lp)->ambient.r = *val;
                        (material+lp)->ambient.g = *(val+1);
                        (material+lp)->ambient.b = *(val+2);
                }
                else if(strcmp(tank, DIFFUSE) == EQUAL){
                        if(l_fg == FALSE){
                                fprintf(stderr,
                                "New Material Label undefined (%s)\n", name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f %f", tank, val, val+1, val+2);
                        (material+lp)->diffuse.r = *val;
                        (material+lp)->diffuse.g = *(val+1);
                        (material+lp)->diffuse.b = *(val+2);
                }
                else if(strcmp(tank, SPECULER) == EQUAL){
                        if(l_fg == FALSE){
                                fprintf(stderr,
                                "New Material Label undefined (%s)\n", name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f %f", tank, val, val+1, val+2);
                        (material+lp)->speculer.r = *val;
                        (material+lp)->speculer.g = *(val+1);
                        (material+lp)->speculer.b = *(val+2);
                }
                else if(strcmp(tank, EXPONENT) == EQUAL){
                        if(l_fg == FALSE){
                                fprintf(stderr,
                                "New Material Label undefined (%s)\n", name);
                                exit(1);
                        }
                        sscanf(line, "%s %f", tank, val);
                        (material+lp)->expo = *val;
                }
                else if(strcmp(tank, LIGHTMODEL) == EQUAL){
                        if(l_fg == FALSE){
                                fprintf(stderr,
                                "New Material Label undefined (%s)\n", name);
                                exit(1);
                        }
                        sscanf(line, "%s %d", tank, &l);
                        if(l < 1)               l = 0;
                        else if(l > 3)          l = 3;
                        (material+lp)->lmodel = l;
                }
                else if(strcmp(tank, SMOOTH) == EQUAL){
                        sscanf(line, "%s %s", tank, tank2);
                        if(strcmp(tank2, "off") == EQUAL)
                                (material+lp)->smooth = OFF;
                        else
                                (material+lp)->smooth = ON;
                }
                else if(strcmp(tank, AMBMAP) == EQUAL){
                        sscanf(line, "%s %s %f %f %f",
```

```
                                        tank, tank2, val, val+1, val+2);
                if(*val < SMOLE || *(val+1) < SMOLE){
                        fprintf(stderr,
                "Mapping time must > 0 ! (%s material=%d\n", name, lp);
                        exit(1);
                }
                strcpy((material+lp)->ambmap.name, tank2);
                (material+lp)->ambmap.time.u = *(val);
                (material+lp)->ambmap.time.v = *(val+1);
                (material+lp)->ambmap.rate = *(val+2);
                (material+lp)->ambmap.mptr = MapFileProcess(line);
                (material+lp)->map = ON;
                (material+lp)->ambmap.flag = ON;
        }
        else if(strcmp(tank, DIFMAP) == EQUAL){
                sscanf(line,
                "%s %s %f %f %f", tank, tank2, val, val+1, val+2);
                if(*val < SMOLE || *(val+1) < SMOLE){
                        fprintf(stderr,
                "Mapping time must > 0 ! (%s material=%d\n", name, lp);
                        exit(1);
                }
                strcpy((material+lp)->difmap.name, tank2);
                (material+lp)->difmap.time.u = *(val);
                (material+lp)->difmap.time.v = *(val+1);
                (material+lp)->difmap.rate = *(val+2);
                (material+lp)->difmap.mptr = MapFileProcess(line);
                (material+lp)->map = ON;
                (material+lp)->difmap.flag = ON;
        }
        else if(strcmp(tank, SPECMAP) == EQUAL){
                sscanf(line, "%s %s %f %f %f",
                                        tank, tank2, val, val+1, val+2);
                if(*val < SMOLE || *(val+1) < SMOLE){
                        fprintf(stderr,
                "Mapping time must > 0 ! (%s material=%d\n", name, lp);
                        exit(1);
                }
                strcpy((material+lp)->specmap.name, tank2);
                (material+lp)->specmap.time.u = *(val);
                (material+lp)->specmap.time.v = *(val+1);
                (material+lp)->specmap.rate = *(val+2);
                (material+lp)->specmap.mptr = MapFileProcess(line);
                (material+lp)->map = ON;
                (material+lp)->specmap.flag = ON;
        }
    }
    MatSum += ln;
    return(ln);
}

int     MapFileProcess(line)
char    *line;
{
        int             i, j, same_fg, mp, ln, lp;
        char            tank[MAXLINE], mapf[MAXLINE];

        sscanf(line, "%s %s", tank, mapf);
        for(i = 0, same_fg = FALSE; i < MapFileNum; i++)
        {
                if(strcmp((mapfile+i)->name, mapf) == EQUAL){
                        same_fg = TRUE;
                        mp = i;
                        break;
                }
        }
        if(same_fg == FALSE){
                mp = MapFileNum;
                ReadMapFile(mp, mapf);
                strcpy((mapfile+mp)->name, mapf);
                MapFileNum++;
        }
        return(mp);
}

/*
*       ***************
*       * ReadMapFile *
*       ***************
*
*       Function: reading mapping file header
*       Usage:
*               int     mp (input) - mapping file number
*               char    *name (input) - mapping file name
*/

ReadMapFile(mp, name)
int     mp;
char    *name;
{
        int     fd, s, k1, k2;
        long    pos;
        U_Char  head[6];
        FILE    *fp;

        fp = fopen(name, "r");
        if(fp == (FILE *)NULL){
                fprintf(stderr, "Can't Open File %s\n", name);
                exit(1);
        }
        fd = fileno(fp);

        s = read(fd, head, HEADER);
        if(s == ERR){
                fprintf(stderr, "Read Error! \n");
                exit(1);
        }

        k1 = *(head+1);
        k2 = *(head+2);
        (mapfile+mp)->size.x = ( k2 << 8) | k1;

        k1 = *(head+3);
        k2 = *(head+4);
        (mapfile+mp)->size.y = ( k2 << 8) | k1;

        (mapfile+mp)->line = (long)((mapfile+mp)->size.x);
        fclose(fp);
}

/*
*       *********************
*       * PrePolygonProcess *
*       *********************
*
*       Function: search polygon conectibility to each vertex &
*                 & calculate each vertex normal vector
*       Usage:
*                 struct Object *op (input) - pointer to object
*/

PrePolygonProcess(op)
struct  Object  *op;
{
        int             i, j, j1;
        int             ep, vp, vn, pn;
        float           sq;
        float           point[MAXEDGE][4];
        Vector3f        vector, sum;

        /* search polygon conectibility to vertex */
        for(i = 0; i < op->ElemNum; i++)
        {
                ep = i + op->eptr;
                vn = (element+ep)->VertNum;
                for(j = 0; j < vn; j++)
                {
                        vp = (element+ep)->vert[j] - 1; /* vertex start from 0*/
                        pn = (vertex+vp)->CntPolyNum;
                        *((vertex+vp)->cntpoly+pn) = ep;
                        (vertex+vp)->CntPolyNum += 1;

                        *(point[j]) = (vertex+vp)->coord.x;
                        *(point[j]+1) = (vertex+vp)->coord.y;
                        *(point[j]+2) = (vertex+vp)->coord.z;
                        *(point[j]+3) = 1.0;
                }
                CalPolygonCenter(&((element+ep)->center), vn, point);
                CalPolygonNormal(&((element+ep)->norm), vn, point);
        }
        /* calculate vertex normal vector */
        for(i = 0; i < op->PointNum; i++)
        {
                vp = i + op->pptr;
                pn = (vertex+vp)->CntPolyNum;
                sum.x = sum.y = sum.z = 0;
                for(j = 0; j < pn; j++)
                {
                        ep = *((vertex+vp)->cntpoly+j);
                        sum.x += (element+ep)->norm.x;
                        sum.y += (element+ep)->norm.y;
                        sum.z += (element+ep)->norm.z;
                }
                vector.x = sum.x / (float)pn;
                vector.y = sum.y / (float)pn;
                vector.z = sum.z / (float)pn;

                UnitVector(&((vertex+vp)->norm), vector);
        }
}

ClearVertex()
{
        int     i;
```

```
        for(i = 0; i < PointSum; i++)
                (vertex+i)->CntPolyNum = 0;
}

/*
*       ************************
*       * CorrectDistortPoint *
*       ************************
*
*       Function: correct distortion between perspective coord. and eye coord.
*       Usage:
*               float   s0 (input) - value on perspective coord. (1~0)
*               Point   *va, *vb (input) - pointer to two terminal points
*               float   s  (output) - value on eye coord. (1~0)
*/

float   CorrectDistortPoint(s0, va, vb)
float   s0;
Point   *va, *vb;
{
        float   s;

        s = -va->coord2.w * s0 / ((vb->coord2.w - va->coord2.w) * s0
                                                        - vb->coord2.w);
        return(s);
}

/*
*       *****************
*       * PersProject2 *
*       *****************
*
*       Function: PersProject ver. 2 (with w)
*       Usage:
*               Point   *po (input) - pointer to point
*               int     pn  (input) - number ofvertice
*/

PersProject2(po, pn)
Point   *po;
int     pn;
{
        int     i, j;
        float   val;
        Point   out[MAXEDGE];

        for(i = 0, j = 0; i < pn; i++)
        {
                if((po+i)->coord.z > 0.0){
                        (out+j)->coord2.x = view.para1 * (po+i)->coord.x /
                                                        (po+i)->coord.z;
                        (out+j)->coord2.y = view.para1 * (po+i)->coord.y /
                                                        (po+i)->coord.z;
                        (out+j)->coord2.z = view.para3 * (1.0 - view.heather /
                                                        (po+i)->coord.z);
                        if(ABS(view.para1) < SMOLE) /* impossible case */
                                val = SMOLE;
                        else
                                val = view.para1;

                        (out+j)->coord2.w = (po+i)->coord.z / val;
                        PointCopyV(out+j, po+i);
                        j++;
                }
        }
        pn = j;
        for(i = 0; i < pn; i++)
        {
                (po+i)->coord2.x = (out+i)->coord2.x;
                (po+i)->coord2.y = (out+i)->coord2.y;
                (po+i)->coord2.z = (out+i)->coord2.z;
                (po+i)->coord2.w = (out+i)->coord2.w;
                PointCopyV(po+i, out+i);
        }
        return(pn);
}

FILE    *fopen();

/*
*       *************************
*       * SmoothPolygonProcess *
*       *************************
*
*       Function: backfacing polygon removal/clipping/perspective transform/
*       rendering with smooth shading and mapping(if they are necessary)
*       Usage:
*               struct Object *op (input) - pointer to object
*/

SmoothPolygonProcess(op)
struct Object *op;
{
        int             i, j, null_fg;
        int             ep, vp, vn, pn, mp, ap, ip;
        float           zval[MAXEDGE];
        float           sq, vis_fg;
        Vector3f        eye, vector;
        Polygon         polygon;
        Illum           illum[MAXLIGHT];
        Edge            edge[MAXEDGE];
        Point           point[MAXEDGE];


        mp = op->mptr;
        if((material+mp)->map == ON){
                map_fg = ON;
                if((material+mp)->ambmap.flag == ON){
                        ambmap_fg = ON;
                        OpenMapFile((material+mp)->ambmap.mptr);
                }
                else
                        ambmap_fg = OFF;
                if((material+mp)->difmap.flag == ON){
                        difmap_fg = ON;
                        OpenMapFile((material+mp)->difmap.mptr);
                }
                else
                        difmap_fg = OFF;
                if((material+mp)->specmap.flag == ON){
                        specmap_fg = ON;
                        OpenMapFile((material+mp)->specmap.mptr);
                }
                else
                        specmap_fg = OFF;
        }
        else
                map_fg = OFF;
        if((material+mp)->smooth == ON)
                smooth_fg = ON;
        else
                smooth_fg = OFF;
        for(i = 0; i < op->ElemNum; i++)
        {
                ep = i + op->eptr;
                vn = (element+ep)->VertNum;
                polygon.norm.x = (element+ep)->norm.x;
                polygon.norm.y = (element+ep)->norm.y;
                polygon.norm.z = (element+ep)->norm.z;

                polygon.center.x = (element+ep)->center.x;
                polygon.center.y = (element+ep)->center.y;
                polygon.center.z = (element+ep)->center.z;

                for(j = 0; j < vn; j++)
                {
                        vp = (element+ep)->vert[j] - 1;/*Element start from 1*/
                        (point+j)->coord.x = (vertex+vp)->coord.x;
                        (point+j)->coord.y = (vertex+vp)->coord.y;
                        (point+j)->coord.z = (vertex+vp)->coord.z;
                        (point+j)->norm.x = (vertex+vp)->norm.x;
                        (point+j)->norm.y = (vertex+vp)->norm.y;
                        (point+j)->norm.z = (vertex+vp)->norm.z;
                        if(map_fg == ON){
                                ap = (element+ep)->mapv[j] - 1;
                                /* Mapvert start from 1*/
                                (point+j)->map.u = (mapvert+ap)->map.u;
                                (point+j)->map.v = (mapvert+ap)->map.v;
                        }
                }

                /* backfacing surface removal */
                vector.x = -(element+ep)->center.x;
                vector.y = -(element+ep)->center.y;
                vector.z = -(element+ep)->center.z;
                null_fg = UnitVector(&eye, vector);
                if(null_fg == 1)
                        eye.z = -1;
                vis_fg = eye.x * (element+ep)->norm.x + eye.y
                        * (element+ep)->norm.y + eye.z * (element+ep)->norm.z;
                if(vis_fg > 0.0) /* visible case */ {
                        polygon.LitNum = LitNum;
                        polygon.mptr = mp;

                        /*********************************/
                        /* clipping at view point pyramid */
                        /*********************************/

                        pn = PolygonClip2(point, vn, 0, 0);
                        pn = PolygonClip2(point, pn, 0, 1);
                        pn = PolygonClip2(point, pn, 1, 0);
                        pn = PolygonClip2(point, pn, 1, 1);
```

```
                        /*********************************/
                        /* perspective transformation    */
                        /*********************************/

                        pn = PersProject2(point, pn);

                        /***  wirte polygon into z-buffer *********/
                        /*  with viewport transformation        */
                        /****************************************/

                        if(pn > 2){
                                /* viewport transformation */
                                for(j = 0; j < pn; j++)
                                {
                                        (point+j)->coord2.x =
                                (point+j)->coord2.x * view.sc2.x + view.sc2.x;
                                        (point+j)->coord2.y =
                                (point+j)->coord2.y * view.sc2.x + view.sc2.y;
                                        /*  fix bug Sep. 9 */
                                }
                                CalPolygonNormal2(&polygon.norm2, pn, point);
                                if(ABS(polygon.norm2.z) >= SMOLE)/* visible */{
                                 CalPolygonCenter2(&polygon.center2,pn, point);
                                        polygon.cn =
                                        polygon.center2.x * polygon.norm2.x
                                        + polygon.center2.y * polygon.norm2.y
                                        + polygon.center2.z * polygon.norm2.z;
                                        polygon.zinc = -polygon.norm2.x
                                                        / polygon.norm2.z;
                                   CalPolygonRegion2(&polygon.valy, pn, point);
                                        CalPolygonEdge2(edge, pn, point);
                                        polygon.EdgeNum = pn;
                                     PolygonScanConvert2(polygon, edge, point);
                                }
                        }
                }
        }
        if(map_fg == ON){
                if(ambmap_fg == ON)
                        CloseMapFile((material+mp)->ambmap.mptr);
                if(difmap_fg == ON)
                        CloseMapFile((material+mp)->difmap.mptr);
                if(specmap_fg == ON)
                        CloseMapFile((material+mp)->specmap.mptr);
        }
}


typedef struct{
                int     flag;
                int     CrossNum;
                int     etable[MAXEDGE]; /* edge table */
                float   ctable[MAXEDGE]; /* x-cross value table */
                int     itable[MAXEDGE]; /* integer x-cross */
        } Cross;

/*
*       **********************
*       * PolygonScanConvert2 *
*       **********************
*
*       Function: scan convert a polygon with vertex atribute
*       Usage:
*               Polygon  polygon (input) - current polygon
*               Edge     *edge    (input) - pointer to edge
*               Point    *point   (input) - pointer to point
*/

PolygonScanConvert2(polygon, edge, point)
Polygon         polygon;
Edge            *edge;
Point           *point;
{
        int             y, k, x, i, sgn, ep, ep0, ep1;
        int             under, up;
        int             left, right, in_flag;
        float           depth, depth0, midr;
        U_Char          ReorderColor();
        Cross           cross;
        Color           color;
        Point           mid[2];


        if(polygon.valy.min < 0) under = 0;
        else                    under = polygon.valy.min;
        if(polygon.valy.max >= view.screen.y)   up = view.screen.y - 1;
        else                                    up = polygon.valy.max;

        for(y = under; y <= up; y++)
        {
                SearchCross(&cross, y, polygon.EdgeNum, edge);
                if(cross.flag != -1){
                        ep = *(cross.etable);
                        depth0 = (polygon.cn - polygon.norm2.y * (float)y) /
                                                        polygon.norm2.z;
                        sgn = (cross.CrossNum + 1) / 2;
                        for(k = 0; k < sgn; k++)
                        {
                                ep0 = 2 * k;
                                ep1 = ep0 + 1;
                                left = *(cross.itable+ep0);
                                right = *(cross.itable+ep1);
                                for(x = left, in_flag = FALSE; x <= right; x++)
                                {
                                        depth = polygon.zinc * (float)x
                                                                + depth0;
                                        if(depth < *(Depth.buf[y]+x)){
                                                if(in_flag == FALSE){
                                                   InBetweenMidAtribute(mid,
&midr, y, edge, point, *(cross.etable+ep0), *(cross.etable+ep1), polygon);
                                                        in_flag = TRUE;
                                                }
                                                InBetweenPixelAtribute(&color,
                                                        x, mid, midr, polygon);
                                                *(Depth.buf[y]+x) = depth;
                                                *(Red.buf[y]+x) =
                                                ReorderColor(color.r);
                                                *(Green.buf[y]+x) =
                                                ReorderColor(color.g);
                                                *(Blue.buf[y]+x) =
                                                ReorderColor(color.b);
                                        }
                                }
                        }
                }
        }
}

ClearMaterial2()
{
        int     i;

        for(i = 0; i < MAXMATERIAL; i++)
        {
                (material+i)->ambient.r = (material+i)->ambient.g
                                        = (material+i)->ambient.b = 0;
                (material+i)->diffuse.r = (material+i)->diffuse.g
                                        = (material+i)->diffuse.b = 0;
                (material+i)->speculer.r = (material+i)->speculer.g
                                        = (material+i)->speculer.b = 0;
                (material+i)->lmodel = 0;
                (material+i)->expo = 1;
                (material+i)->smooth = OFF; /* add for ver. 2  8/25 */
                (material+i)->map = OFF;
                (material+i)->ambmap.flag = OFF;
                (material+i)->difmap.flag = OFF;
                (material+i)->specmap.flag = OFF;
        }
}


/*
*       ****************
*       * PolygonClip2 *
*       ****************
*
*       Function: polygon clipper with vartex atribute
*       Usage:
*               Point    *point    (input) - pointer to point
*               int      vn        (input) - number of point
*               int      fg1, fg2(input) - clipping flags
*/

PolygonClip2(point, vn, fg1, fg2)
Point   *point;
int     vn;
int     fg1, fg2;
{
        int     i, i1;
        int     pn, bio, pio, sio;
        float   v0, z0, ag0;
        float   pnt[4], clpm[MAXEDGE][4];
        Point   mp, vert[MAXEDGE];


        if(fg1 == 0)            ag0 = 1;
        else                    ag0 = -1;

        for(i = 0; i < vn; i++)
        {
                *pnt = (point+i)->coord.x;
                *(pnt+1) = (point+i)->coord.y;
                *(pnt+2) = (point+i)->coord.z;
                *(pnt+3) = 1.0;
```

```
                CalMatV(clpm[i], pnt, ClipMat);
        }

        pn = 0;
        v0 = ag0 * clpm[0][fg2];
        z0 = -1.0 * clpm[0][2];

        if(v0 < z0)
                sio = 0;
        else{
                sio = 1;
                PointCopyV(vert, point);
                pn++;
        }
        bio = sio;
        for(i = 1; i < vn; i++)
        {
                i1 = i - 1;
                v0 = ag0 * clpm[i][fg2];
                z0 = -1.0 * clpm[i][2];

                if(v0 < z0){
                        pio = 0;
                        if(bio == 1){
                                CalCrossWall2(point+i, point+i1, &mp, fg1, fg2);
                                PointCopyV(vert+pn, &mp);
                                pn++;
                        }
                }
                else{
                        pio = 1;
                        if(bio == 1){
                                PointCopyV(vert+pn, point+i);
                                pn++;
                        }
                        else{
                                CalCrossWall2(point+i, point+i1, &mp, fg1, fg2);
                                PointCopyV(vert+pn, &mp);
                                pn++;

                                PointCopyV(vert+pn, point+i);
                                pn++;
                        }
                }
                bio = pio;
        }
        i1 = vn - 1;
        if((bio == 0 && sio == 1) || (bio == 1 && sio == 0)){
                CalCrossWall2(point, point+i1, &mp, fg1, fg2);
                PointCopyV(vert+pn, &mp);
                pn++;
        }
        for(i = 0; i < pn; i++)
                PointCopyV(point+i, vert+i);
        return(pn);
}

CalCrossWall2(vt, vm, mp, fg1, fg2)
Point   *vt, *vm, *mp;
int     fg1, fg2;
{
        float   clp, t, ak, bk, ck, z;
        float   xrg, yrg, zrg;


        if(fg2 == 0){
                if(fg1 == 0)            clp = view.para1;
                else                    clp = -1.0 * view.para1;

                ck = vm->coord.z - vt->coord.z;

                if(ABS(ck) >= SMOLE){
                        bk = -1.0 * clp * (vm->coord.x - vt->coord.x);
                        zrg = ck - bk;
                        if(ABS(zrg) >= SMOLE){
                                ak = -1.0 * clp * vt->coord.x;
                                z = (ak * ck - vt->coord.z * bk) / zrg;
                                t = (z - vt->coord.z) / ck;
                                InBetweenPointAtribute(mp, vt, vm, t);
                                mp->coord.x = vt->coord.x + t
                                                * (vm->coord.x - vt->coord.x);
                                mp->coord.y = vt->coord.y + t
                                                * (vm->coord.y - vt->coord.y);
                                mp->coord.z = z;
                        }
                        else
                                PointCopyV(mp, vm); /*impossible case */
                }
                else
                {
                        mp->coord.z = vt->coord.z;
                        mp->coord.x = -1.0 * vt->coord.z / clp;
                        xrg = vm->coord.x - vt->coord.x;
                        if(ABS(xrg) >= SMOLE){
                                t = (mp->coord.x - vt->coord.x) / xrg;
                                mp->coord.y = vt->coord.y + t
                                                * (vm->coord.y - vt->coord.y);
                                InBetweenPointAtribute(mp, vt, vm, t);
                        }
                        else /* imposible case */
                        {
                                mp->coord.y = vt->coord.y;
                                mp->norm.x = vt->norm.x;
                                mp->norm.y = vt->norm.y;
                                mp->norm.z = vt->norm.z;
                                if(map_fg == ON){
                                        mp->map.u = vt->map.u;
                                        mp->map.v = vt->map.v;
                                }
                        }
                }
        }
        else{
                if(fg1 == 0)            clp = view.para2;
                else                    clp = -1.0 * view.para2;

                ck = vm->coord.z - vt->coord.z;

                if(ABS(ck) >= SMOLE){
                        bk = -1.0 * clp * (vm->coord.y - vt->coord.y);
                        zrg = ck - bk;
                        if(ABS(zrg) >= SMOLE){
                                ak = -1.0 * clp * vt->coord.y;
                                z = (ak * ck - vt->coord.z * bk) / zrg;
                                t = (z - vt->coord.z) / ck;
                                InBetweenPointAtribute(mp, vt, vm, t);
                                mp->coord.x = vt->coord.x + t
                                                * (vm->coord.x - vt->coord.x);
                                mp->coord.y = vt->coord.y + t
                                                * (vm->coord.y - vt->coord.y);
                                mp->coord.z = z;
                        }
                        else
                                PointCopyV(mp, vm); /*impossible case */
                }
                else{
                        mp->coord.z = vt->coord.z;
                        mp->coord.y = -1.0 * vt->coord.z / clp;
                        yrg = vm->coord.y - vt->coord.y;
                        if(ABS(yrg) >= SMOLE){
                                t = (mp->coord.y - vt->coord.y) / yrg;
                                mp->coord.x = vt->coord.y + t
                                                * (vm->coord.x - vt->coord.x);
                                InBetweenPointAtribute(mp, vt, vm, t);
                        }
                        else /* imposible case */ {
                                mp->coord.x = vt->coord.x;
                                mp->norm.x = vt->n
                                mp->norm.y = vt->no
                                mp->norm.z = vt->norm..
                                if(map_fg == ON){
                                        mp->map.u = vt->map.u;
                                        mp->map.v = vt->map.v;
                                }
                        }
                }
        }
}

/*
*       *************************
*       * InBetweenPointAtribute *
*       *************************
*
*       Function: inbetween point atribute
*       Usage:
*               Point   *vt, *vm (input) - two terminal point of one line
*               Point   *mp     (input) - inbetween point
*               float   t               (input) - inbetween value
*/

InBetweenPointAtribute(mp, vt, vm, t)
Point   *mp, *vt, *vm;
float   t;
{
        Vector3f            vector;

        vector.x = vt->norm.x + (vm->norm.x - vt->norm.x) * t;
        vector.y = vt->norm.y + (vm->norm.y - vt->norm.y) * t;
        vector.z = vt->norm.z + (vm->norm.z - vt->norm.z) * t;
        UnitVector(&(mp->norm), vector);
```

```
        if(map_fg == ON){
                mp->map.u = vt->map.u + (vm->map.u - vt->map.u) * t;
                mp->map.v = vt->map.v + (vm->map.v - vt->map.v) * t;
        }
}


PointCopyV(copy, orig)
Point   *copy, *orig;
{
        copy->coord.x = orig->coord.x;
        copy->coord.y = orig->coord.y;
        copy->coord.z = orig->coord.z;

        copy->norm.x = orig->norm.x;
        copy->norm.y = orig->norm.y;
        copy->norm.z = orig->norm.z;

        copy->color.r = orig->color.r;
        copy->color.g = orig->color.g;
        copy->color.b = orig->color.b;

        copy->map.u = orig->map.u;
        copy->map.v = orig->map.v;
}

CalPolygonEdge2(edge, pn, point)
Edge    *edge;
int     pn;
Point   *point;
{
        int     j, j1;
        float   ay;

        for(j = 0; j < pn; j++)
        {
                j1 = j + 1;
                if(j1 == pn)            j1 = 0;

                ay = (point+j1)->coord2.y - (point+j)->coord2.y;
                if(ABS(ay) < SMOLE)             (edge+j)->flag = -1;
                else{
                        (edge+j)->flag = 1;
                        (edge+j)->cline = ((point+j1)->coord2.x
                                                - (point+j)->coord2.x) / ay;
                }
                (edge+j)->posi.x = (point+j)->coord2.x;
                (edge+j)->posi.y = (point+j)->coord2.y;
                (edge+j)->posi.z = (point+j)->coord2.z;
                (edge+j)->valy.min = MIN((point+j)->coord2.y,
                                                        (point+j1)->coord2.y);
                (edge+j)->valy.max = MAX((point+j)->coord2.y,
                                                        (point+j1)->coord2.y);
                (edge+j)->vp0 = j;
                (edge+j)->vp1 = j1;
        }
}

CalPolygonRegion2(valy, vn, point)
Minmax  *valy;
int     vn;
Point   *point;
{
        int     i;

        valy->min = valy->max = point->coord2.y;
        for(i = 1; i < vn; i++)
        {
                valy->min = MIN(valy->min, (point+i)->coord2.y);
                valy->max = MAX(valy->max, (point+i)->coord2.y);
        }
}


CalPolygonCenter2(center, vn, point)
Vector3f        *center;
int             vn;
Point           *point;
{
        int             i;
        Vector3f        sum;

        sum.x = sum.y = sum.z = 0;
        for(i = 0; i < vn; i++)
        {
                sum.x += (point+i)->coord2.x;
                sum.y += (point+i)->coord2.y;
                sum.z += (point+i)->coord2.z;
        }
        center->x = sum.x / (float)vn;
        center->y = sum.y / (float)vn;
        center->z = sum.z / (float)vn;
}

CalPolygonNormal2(norm, vn, point)
Vector3f        *norm;
int             vn;
Point           *point;
{
        int             i, j, vn1;
        float           sq;
        Vector3f        sum;

        sum.x = sum.y = sum.z = 0;
        vn1 = vn - 1;
        for(i = 0; i < vn; i++)
        {
                if(i == vn1)    j = 0;
                else            j = i + 1;
                sum.x += ((point+i)->coord2.y - (point+j)->coord2.y)
                                * ((point+i)->coord2.z + (point+j)->coord2.z);
                sum.y += ((point+i)->coord2.z - (point+j)->coord2.z)
                                * ((point+i)->coord2.x + (point+j)->coord2.x);
                sum.z += ((point+i)->coord2.x - (point+j)->coord2.x)
                                * ((point+i)->coord2.y + (point+j)->coord2.y);
        }
        UnitVector(norm, sum);
        /* positive = clockwize */
        norm->x *= -1.0;
        norm->y *= -1.0;
        norm->z *= -1.0;
}

/*
*       ************************
*       * InBetweenMidAtribute *
*       ************************
*
*       Function: inbetween two points with Y-direction
*       Usage:
*               Point   *mid (input) - pointer to inbetween point
*               float   *midr (input) - inbetween value
*               int     y    (input) - screen Y value
*               Edge    *edge (input) - pointer to edge
*               Point   *point(input) - pointer to point
*               int     et0, et1 (input) - the numbers of edge of current line
*/

InBetweenMidAtribute(mid, midr, y, edge, point, et0, et1, polygon)
Point   *mid;
float   *midr;
int     y;
Edge    *edge;
Point   *point;
int     et0, et1;
Polygon polygon;
{
        float           sq, s, t, s1, t1, r1, s0, t0, r0;
        int             i, vp0, vp1, vq0, vq1;
        float           CorrectDistortPoint();


        vp0 = (edge+et0)->vp0;
        vp1 = (edge+et0)->vp1;
        vq0 = (edge+et1)->vp0;
        vq1 = (edge+et1)->vp1;

        s = (point+vp1)->coord2.y - (point+vp0)->coord2.y;
        if(ABS(s) >= SMOLE)     s0 = (y - (point+vp0)->coord2.y) / s;
        else                    s0 = 0;
        s1 = CorrectDistortPoint(s0, point+vp0, point+vp1);
        LinearInBetween(mid, point+vp0, point+vp1, s1, polygon);

        t = (point+vq1)->coord2.y - (point+vq0)->coord2.y;
        if(ABS(t) >= SMOLE)     t0 = (y - (point+vq0)->coord2.y) / t;
        else                    t0 = 0;
        t1 = CorrectDistortPoint(t0, point+vq0, point+vq1);
        LinearInBetween(mid+1, point+vq0, point+vq1, t1, polygon);

        PersProject2(mid, 2);
        for(i = 0; i < 2; i++)
                (mid+i)->coord2.x = (mid+i)->coord2.x * view.sc2.x + view.sc2.x;
        *midr = (mid+1)->coord2.x - mid->coord2.x;
}

/*
*       *************************
*       * InBetweenPixelAtribute *
*       *************************
*
*       Function: inbetween two points with X-direction
*       Usage:
*               Color   *color (input) - pointer to color
*               Point   *mid (input) - pointer to inbetween point
```

```
*                   float       *midr (input) - inbetween value
*                   int         x    (input) - screen X value
*                   Polygon polygon (input) - current polygon
*/

InBetweenPixelAtribute(color, x, mid, midr, polygon)
Color     *color;
int       x;
Point     *mid;
float     midr;
Polygon   polygon;
{
          float               r0, r1;
          Point               hit;
          Vector3f            eye;
          Illum               illum[MAXLIGHT];
          float               CorrectDistortPoint();
          if(ABS(midr) >= SMOLE)     r0 = ((float)x - mid->coord2.x) / midr;
          else                       r0 = 0;
          r1 = CorrectDistortPoint(r0, mid, mid+1);
          LinearInBetween(&hit, mid, mid+1, r1, polygon);

          polygon.center.x = hit.coord.x;
          polygon.center.y = hit.coord.y;
          polygon.center.z = hit.coord.z; /*for using CalPolygonIllum()*/
          CalPolygonIllum(illum, polygon);

          UnitVector(&eye, hit.coord);
          eye.x *= -1.0;
          eye.y *= -1.0;
          eye.z *= -1.0;

          CalPixelColor(color, polygon, hit, illum, eye);
}

/*
*         ********************
*         * LinearInBetween *
*         ********************
*
*         Function: Inbetween two point atribute
*         Usage:
*                   Point     *mid (input) - inbetween point
*                   Point     *ed0, *ed1 (input) - two terminal point
*                   float     r            (input) - inbetween value
*                   Polygon polygon    (input) - current polygon
*/

LinearInBetween(mid, ed0, ed1, r, polygon)
Point     *mid, *ed0, *ed1;
float     r;
Polygon   polygon;
{
          Vector3f            vector;

          mid->coord.x = (ed1->coord.x - ed0->coord.x) * r + ed0->coord.x;
          mid->coord.y = (ed1->coord.y - ed0->coord.y) * r + ed0->coord.y;
          mid->coord.z = (ed1->coord.z - ed0->coord.z) * r + ed0->coord.z;

          if(smooth_fg == ON){
                    vector.x = (ed1->norm.x - ed0->norm.x) * r + ed0->norm.x;
                    vector.y = (ed1->norm.y - ed0->norm.y) * r + ed0->norm.y;
                    vector.z = (ed1->norm.z - ed0->norm.z) * r + ed0->norm.z;
                    UnitVector(&(mid->norm), vector);
          }
          else{
                    mid->norm.x = polygon.norm.x;
                    mid->norm.y = polygon.norm.y;
                    mid->norm.z = polygon.norm.z;
          }
          if(map_fg == ON){
                    mid->map.u = (ed1->map.u - ed0->map.u) * r + ed0->map.u;
                    mid->map.v = (ed1->map.v - ed0->map.v) * r + ed0->map.v;
          }
}

UnitVector(vector, orig)
Vector3f            *vector, orig;
{
          float               sq;
          int                 flag;

          sq =sqrt((double)(orig.x * orig.x + orig.y * orig.y + orig.z * orig.z));
          if(ABS(sq) >= SMOLE){
                    vector->x = orig.x / sq;
                    vector->y = orig.y / sq;
                    vector->z = orig.z / sq;
                    flag = 0;
          }
          else{
                    vector->x = 0;
                    vector->y = 0;
                    vector->z = 1;
                    flag = 1;
          }
          return(flag);
}

/*
*         *****************
*         * CalPixelColor *
*         *****************
*
*         Function: calulate pixel color with mapping color
*         Usage:
*                   Color     *color     (input) - pointer to color
*                   Polygon polygon (input) - current polygon
*                   Illum     *illum  (input) - pointer to illum
*                   Vector3f eye    (input) - eye vector (to current dot)
*/

CalPixelColor(color, polygon, hit, illum, eye)
Color               *color;
Polygon             polygon;
Point               hit;
Illum               *illum;
Vector3f            eye;
{
          int       j, mptr;
          float     shading, reflect, bright;
          float     sq;
          float     rate, urate;
          float     Shading(), Speculer();
          Color     bri, hi, lpw, ambm, difm, specm;


          bri.r = bri.g = bri.b = 0;
          hi.r = hi.g = hi.b = 0;
          for(j = 0; j < polygon.LitNum; j++)
          {
                    shading = Shading((illum+j)->vec, hit.norm);
                    mptr = polygon.mptr;
                    if((material+mptr)->lmodel > 1)
                              reflect = Speculer((illum+j)->vec, hit.norm, eye,
                                                                  (material+mptr)->expo);

                    lpw.r = (light+j)->color.r * (illum+j)->dump;
                    lpw.g = (light+j)->color.g * (illum+j)->dump;
                    lpw.b = (light+j)->color.b * (illum+j)->dump;

                    bri.r += lpw.r * shading;
                    bri.g += lpw.g * shading;
                    bri.b += lpw.b * shading;

                    if((material+mptr)->lmodel > 1){
                              hi.r += reflect * lpw.r;
                              hi.g += reflect * lpw.g;
                              hi.b += reflect * lpw.b;
                    }
          }
          if(map_fg == OFF){
                    color->r = (material+mptr)->ambient.r
                                        + (material+mptr)->diffuse.r * bri.r;
                    color->g = (material+mptr)->ambient.g
                                        + (material+mptr)->diffuse.g * bri.g;
                    color->b = (material+mptr)->ambient.b
                                        + (material+mptr)->diffuse.b * bri.b;
                    if((material+mptr)->lmodel > 1){
                              color->r += (material+mptr)->speculer.r * hi.r;
                              color->g += (material+mptr)->speculer.g * hi.g;
                              color->b += (material+mptr)->speculer.b * hi.b;
                    }
          }
          else{
                    if(ambmap_fg == ON){
                              CalMapColor(&ambm, (material+mptr)->ambmap.mptr,
                                                  (material+mptr)->ambmap.time, hit);
                              rate = (material+mptr)->ambmap.rate;
                              urate = 1.0 - rate;
                              color->r = (material+mptr)->ambient.r * urate
                                                                  + ambm.r * rate;
                              color->g = (material+mptr)->ambient.g * urate
                                                                  + ambm.g * rate;
                              color->b = (material+mptr)->ambient.b * urate
                                                                  + ambm.b * rate;
                    }
                    else{
                              color->r = (material+mptr)->ambient.r;
                              color->g = (material+mptr)->ambient.g;
                              color->b = (material+mptr)->ambient.b;
                    }
                    if(difmap_fg == ON){
                              if(CompareMap((material+mptr)->ambmap,
                                                  (material+mptr)->difmap) == TRUE)
                                        ColorCopy(&difm, ambm);
```

```
                        else
                                CalMapColor(&difm, (material+mptr)->difmap.mptr,
                                        (material+mptr)->difmap.time, hit);
                        rate = (material+mptr)->difmap.rate;
                        urate = 1.0 - rate;

                        color->r += ((material+mptr)->diffuse.r * urate
                                                + difm.r * rate) * bri.r;
                        color->g += ((material+mptr)->diffuse.g * urate
                                                + difm.g * rate) * bri.g;
                        color->b += ((material+mptr)->diffuse.b * urate
                                                + difm.b * rate) * bri.b;
                }
                else{
                        color->r += (material+mptr)->diffuse.r * bri.r;
                        color->g += (material+mptr)->diffuse.g * bri.g;
                        color->b += (material+mptr)->diffuse.b * bri.b;
                }
                if((material+mptr)->lmodel > 1){
                        if(specmap_fg == ON){
                                if(CompareMap((material+mptr)->ambmap,
                                        (material+mptr)->specmap) == TRUE)
                                        ColorCopy(&specm, ambm);
                                else if(CompareMap((material+mptr)->difmap,
                                        (material+mptr)->specmap) == TRUE)
                                        ColorCopy(&specm, difm);
                                else
                                        CalMapColor(&specm,
        (material+mptr)->specmap.mptr, (material+mptr)->specmap.time, hit);
                                rate = (material+mptr)->specmap.rate;
                                urate = 1.0 - rate;

                                color->r += ((material+mptr)->speculer.r* urate
                                                + specm.r * rate) * hi.r;
                                color->g += ((material+mptr)->speculer.g* urate
                                                + specm.g * rate) * hi.g;
                                color->b += ((material+mptr)->speculer.b* urate
                                                + specm.b * rate) * hi.b;
                        }
                        else{
                                color->r += (material+mptr)->speculer.r * hi.r;
                                color->g += (material+mptr)->speculer.g * hi.g;
                                color->b += (material+mptr)->speculer.b * hi.b;
                        }
                }
        }
}

CompareMap(comp1, comp2)
Mapping comp1, comp2;
{
        int     flag = FALSE;

        if(comp1.flag == ON){
                if(strcmp(comp1.name, comp2.name) == EQUAL){
                        if(comp1.time.u == comp2.time.u){
                                if(comp1.time.v == comp2.time.v)
                                        flag = TRUE;
                        }
                }
        }
        return(flag);
}

ColorCopy(copy, orig)
Color   *copy, orig;
{
        copy->r = orig.r;
        copy->g = orig.g;
        copy->b = orig.b;
}

OpenMapFile(ip)
int     ip;
{
        long    pos;

        (mapfile+ip)->fp = fopen((mapfile+ip)->name, "r");
        if((mapfile+ip)->fp == (FILE *)NULL){
                fprintf(stderr, "Can't Open File %s\n",         (mapfile+ip)->name);
                exit(1);
        }
        (mapfile+ip)->fd = fileno((mapfile+ip)->fp);
        (mapfile+ip)->posi.x = 0;
        (mapfile+ip)->posi.y = 0;
        (mapfile+ip)->pos = (long)HEADER + 2L * (mapfile+ip)->line;
        pos = lseek((mapfile+ip)->fd, (mapfile+ip)->pos, 0);
        if(pos == -1L){
                fprintf(stderr, "Seek Error !! (OpenMapFile pos=%ld)\n", pos);
                exit(1);
        }
}

CloseMapFile(ip)
int     ip;
{
        fclose((mapfile+ip)->fp);
}

/*
*       ***************
*       * CalMapColor *
*       ***************
*
*       Function: calculate mapping color
*       Usage:
*               Color     *color (input) - pointer to color
*               Vector2f time  (input) - turn of mapping
*               int       ip    (input) - No. of mapfile
*               Point     hit         (input) - current hitting point
*/

CalMapColor(color, ip, time, hit)
Color           *color;
Vector2f        time;
int             ip;
Point           hit;
{
        int     x, y, col;
        U_Char  tank[1];
        long    rel, pos;

        x = hit.map.u * ((mapfile+ip)->size.x - 1) * time.u;
        y = hit.map.v * ((mapfile+ip)->size.y - 1) * time.v;
        if(x < 0)                               x = 0;
        if(x >= (mapfile+ip)->size.x)           x = x % (mapfile+ip)->size.x;
        if(y < 0)                               y = 0;
        if(y >= (mapfile+ip)->size.y)           y = y % (mapfile+ip)->size.y;
        rel = (3L * ((long)y - (long)(mapfile+ip)->posi.y) - 2L)
        * (mapfile+ip)->line + ((long)x - (long)(mapfile+ip)->posi.x) - 1L;
        pos = lseek((mapfile+ip)->fd, rel, 1);
        if(pos == -1L){
                fprintf(stderr,
"Seek Error!(CalMapColor pos=%ld map=%d new:x=%d y=%d old:x=%d y=%d rel=%ld)\n",
                pos, ip, x, y, (mapfile+ip)->posi.x, (mapfile+ip)->posi.y, rel);
                exit(1);
        }
        read((mapfile+ip)->fd, tank, 1);
        col = (int)(*tank);
        color->r = col / (float)BYTE;
        pos = lseek((mapfile+ip)->fd, rel, 1);
        if(pos == -1L){
                fprintf(stderr,
                "Seek Error !!(CalMapColor pos=%ld map=%d x=%d y=%d)\n", pos, ip, x, y);
                exit(1);
        }
        read((mapfile+ip)->fd, tank, 1);
        col = (int)(*tank);
        color->g = col / (float)BYTE;

        rel = (mapfile+ip)->line - 1L;
        pos = lseek((mapfile+ip)->fd, rel, 1);
        if(pos == -1L){
                fprintf(stderr,
                "Seek Error !!(CalMapColor pos=%ld map=%d x=%d y=%d)\n", pos, ip, x, y);
                exit(1);
        }
        read((mapfile+ip)->fd, tank, 1);
        col = (int)(*tank);
        color->b = col / (float)BYTE;

        (mapfile+ip)->posi.x = x;
        (mapfile+ip)->posi.y = y;
}
```

連載 GIS総論(3)

米国におけるGISの現状

太田和　亨*

はじめに

前回はGIS発展の動向を約四半世紀にわたって解説した。本稿を投函した直後にある専門誌（American Cartographer,1988年7月号）が手元に届いたが，皮肉にもGISの発展史を特集していた。GIS史の主人公ともいえる人たちが執筆しているので，ぜひ一読をお勧めする。GIS史に関しては，歴史が浅いためかあまり公表されていない面が多く，GIS発展史の研究に携わる人たちにとってこの特集号は貴重な文献となろう。続いて本稿ではGISの現状がどうなっているのか，探ってみたい。

米国のGIS昨今

GISといえば4,5年前までは一握の専門家に知られていたにすぎなかったが，最近になって市場が急速に拡大し，内容が詳しく把握されているかどうかはともかく，一般にも広く知られるようになってきた。最近の「Computer Graphics Today」誌に載った米国データクエスト社の調査によれば，1988年のGIS市場は2億9,600万ドルと推定されている。前調査の推定では1億9,200万ドルであったから，なんと35%も予想を上回ったことになる。また，この数字を見てもいかにGISの市場が急転しているかがおわかりになろう。各方面の現場の声を聞いてみると，この傾向に今後ますます加速がかかると予測される（**図1**，次ページ）。最近のGIS関連の学会などの参加者の統計を見てみると，現状ではほぼ60～70%が政府，地方自治体関係者またはユーザーであるが，最近の傾向として民間企業やコンサルタントの数が急増していることが注目される。これは，GISの価格低減などの要因もあって，民間企業もようやく最近になってGISの利用価値を見いだし始めたこととして受け取れよう。「GIS World」誌の8月号に掲載されている人事募集欄では50%強が民間企業からのものであり，この事実を裏付けるものであろう。

民間企業の増大にも深く関係がある傾向かもしれないが，元来AM/FMあるいはCADとして開発されたソフトウエアにGIS特有の機能を付加して，GIS市場に参入する企業が急増している現象も注目に値しよう。これを全体面からとらえれば，GISの定義そのものが薄弱化し，今まで明確にあったAM/FMまたはCADとの境界が消滅しかけているものとも受け取れ，その半面，新しい意味でのGIS市場が拡大しているとも読み取れる。しかし当初にも述べたように，真のGISにはマッピング，データベース検索，解析の3機能があるべきで，最近のGISには，GISとよぶにはふさわしくないシステムも出現しているので注意が必要である。

ハードウエア，ソフトウエア面をみてみると，GISもCAD，AM/FM市場と同様ワークステーション時代に入っており，Sun，DEC，HPといったプラットフォームが普及し始めている。オペレーティング・システムの方もCMS，VMSといったメーカー独自のものからUNIXへと，バージョンは異なるにしても一応統一へと向かっている。ソフトウエアについても，当然このようなシステム上で動くものが有望になってくると考えられる。一方，パーソナルコンピュータも新時代に移りつつあり，一昔前のミニコンピュータの演算能力をしのぐ機種が大半になってきているため，汎用というメリットを考慮すれば，これからますますGISの世界で利用されていくことが予想される。

米国GIS市場の裏表

さて，ここでGIS関連の中枢的存在となっている企業をみてみよう。前回にもあげたが，GISの先端を走るのはまぎれもなくロサンゼルス近郊に本社のあるESRIであろう。ESRIのGISソフトウエアARC/INFOは，日本でもパスコから発売されているので一部の読者にはおなじみと思うが，1987年度の数字ではGIS市場の50～60%程度のシェアをもっており，今後市場がさらに拡大するにつれARC/INFOのマーケット・シェアも増大し，応用分野によっては60～80%にもなることが予想される。前述の「Computer Graphics Today」誌によれば，1987年度の同社の売上高は約3,000万ドルであったといわれる。ARC/INFOはこれまでプライム，DEC，データゼネラル，IBMのプラットフォーム上で動いていたが，最近からSunのUNIX上でも走るようになり，現在Sunのワークステーションは同社の推奨ハードウエアともなっている。現在HP，Apolloなどのワークステーションにも移植中で，1988年の末頃までには発売にいたるようである。ARC/INFOの特徴としては，データにトポロジの考え方が導入されており，機能的には優れているがアルゴリズムの効率に疑問の多い人も少なくはない。筆者の経験からいえば，入力用プログラム（ADS）は他のシステムに比べて劣るようである。Ver.4.0ではAML（Arc Macro Language）

*おおたわ　とおる　アイダホ大学準教授　GIS研究室長　College of Art & Architecture, University of Idaho, Moscow, Idaho 83843 U.S.A.

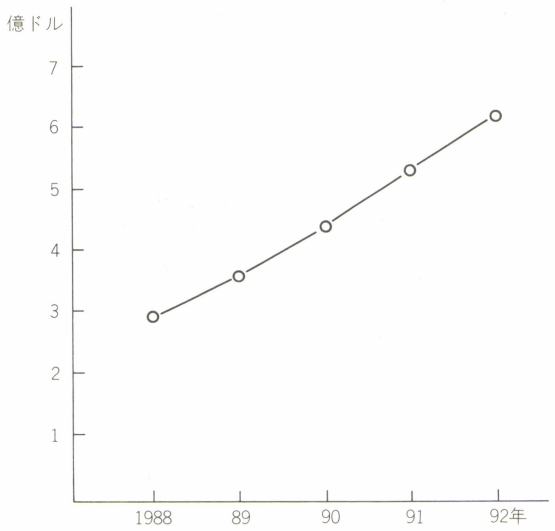


図1　米国における今後5年間のGIS市場の伸び予測
（資料：データクエスト社，Computer Graphics Today 誌による）

やプルダウン・メニューが使えるようになり，ユーザー・インタフェースの質が向上したものの，複雑なシステムのために使いこなせるまでには長期の訓練が必要である。同クラスのシステムにも共通していえることだが，教育用のシステムとしては上級向きで，入門用のGISとしては不適である。

続いてESRIを追い上げるのは，日本にも支社のあるインターグラフ社である。同社の市場は土地の登記や道路施設管理といったAM/FMの世界，あるいは建築土木方面のCADといった分野が主体であるが，80年代前半からGISの機能を追加し始め，現在はDECのVAXのプラットフォームで動いている。インターグラフGISはVAX上でのみ動作することが弱点であるが，CADの世界から発展したGISだけに優れた入力用プログラムを備えているのが特徴である。特に先進のスキャナ技術を持ち合わせているので，コストさえ下がれば一層普及するだろう。1986年時点のGISの売上高は1,500万ドルとみられている。

IBMは売上高では第3位にランクされており，同社のメインフレーム上で動くGFISを発売しているが，1986年には1,000万ドルの売上げがあったとみられている。これは純粋なGISとは少々異なるが，電力会社などの施設管理面で売上げを伸ばしている。

以下テキサス州に本部のあるシナコムテクノロジー社とカナダのジオビジョン社が4位の座を競っているが，ともにVAXベースのGISである。前者はAM/FM分野で多く使用されており，日本でもNCG社が販売を担当している。シナコムのGIS機能は，元来はEMIS（Environmental Management Information System）にあったが，同社のリレーショナル・データベースINFORMAPに統合された。EMISは，前回にも取り上げたハーバード大学LCGSAのODYSSEYの市販版であるので，あえていえばARC/INFOとは遠縁関係にあるものともいえる。一方，ジオビジョン社は長い間米国以外の市場を開拓してきたが，最近，米国市場でも売上げを伸ばしている。

以上，米国市場における主要GISをあげたが，ここで忘れてはならないGISに「GRASS（Geographic Resource Analysis Support System）」がある。GRASSは80年代前半に陸軍工兵隊の土木工学研究所・環境部門において開発されたラスター・システムで，いくつかの連邦政府機関が利用しているが，最近になってベクター機能も使えるようになり有望視されている。GRASSはSun，MASSCOMPのUNIX，DECのVMS上で動き，現在他のプラットフォーム上へと移植されているもようである。C言語で書かれているので移植が容易であり，ソースコードがパブリック・ドメインなので比較的安価に入手できるという利点がある。それゆえ，目的に合ったシステムが構成できる点もメリットであるし，ラスター・システムであるのでランドサット，スポットなどの衛星データ解析にも最適である。

以上，ワークステーションをベースとしたGISを取り上げてみた。最近になってパーソナルコンピュータをプラットフォームにした優秀なソフトウエアが次々と開発されてきているので，主要なものを紹介したいが，残念ながら誌面の都合で別の機会に譲ることとしたい。

国立地域情報・解析センター（National Center for Geographic Information & Analysis：NCGIA）

日本国内でも既知と思うが，米国科学財団（NSF）の援助によってNCGIAという組織の設立が正式に決定した。NCGIAというのは，米国の大学を何校か選んでGIS研究・教育の拠点とし，この先1995年度まで総額1,000万ドルを費やす計画である。設立過程については詳しく書かれている記事があるので参照されたい[1]。NCGIAは，GIS関係者からの長年にわたる設立要請にこたえてこのほど最終選考が終了し，中途で応募全校から数校が選出された後，1988年6月ウィスコンシン大学とカリフォルニア大学サンタバーバラ校（UCSB）の2校が候補校として残ったが，1988年8月19日最終決定がなされ，UCSB校に決まった。ニューヨーク州立大学バッファロー校とメイン大学オロノ校もUCSBのコンソーシアムとなっているので，研究費を分かち合うことになる。この決定によって米国大学のGIS関連プログラムはおおいに動揺しているのが現状である。NCGIAに関しては，日本においても一種のモデルとして注目していったらよいだろう。

おわりに

今回は最新GISソフトウエアとNCGIAについて述べたが，現状といえば他にも数限りないトピックがあろう。残念ながら誌面の都合でこの2点に限られてしまったが，前述したように，パーソナルコンピュータ版GISソフトウエアについては別の機会に紹介するとして，次回はGISの今後の課題について追っていくことにする。

参考文献

1) A. F. Ronald：“The National Science Foundation National Center for Geographic Information and Analysis”, International Journal of Geographical Information Systems, Vol. 1, No. 4, pp. 303-326, 1987. 10-12
2) American Cartographer, 1988. 7
3) GIS World, Vol. 1, No. 2, p. 18, 1988. 9
4) L. Lang：“Surge expected in GIS Growth”, Computer Graphics Today, Vol. 5, No. 8, pp. 1, 15-16, 1988. 8

レイ・トレーシング(5)

曲面パッチのレイ・トレーシングを行うための道具をそろえるとともに，光線と曲面との交点を求めるKajiyaの方法について考える。

大野　義夫*

はじめに

前回(88年9月号)は，代表的な曲面パッチである双3次パッチとBézierパッチについて，その定義を示し，また例を与えた。前回にも述べたように，この2種類の曲面パッチは相互に容易に変換でき，表現形式が異なるだけで，実質的には同じものであるといって差し支えない。今回の説明では，この2つの曲面パッチの定義が非常に重要であるので，まずこれを簡単に復習しておこう。

●双3次パッチ

2つのパラメータ u, v をもつ双3次パッチ $\boldsymbol{G}(u, v)$, $0 \leq u \leq 1$, $0 \leq v \leq 1$ は，曲面の4隅での位置 $\boldsymbol{P}_{ij}$, u 方向の傾き $\boldsymbol{R}_{ij}$, v 方向の傾き $\boldsymbol{S}_{ij}$, それにねじれ $\boldsymbol{T}_{ij}$ に基づき，次のように定められる曲面である($i=0, 1$；$j=0, 1$)。

$$\boldsymbol{G}(u, v)=[h_0(u)\ h_1(u)\ \overline{h}_0(u)\ \overline{h}_1(u)] \times \begin{bmatrix} \boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{S}_{00} & \boldsymbol{S}_{01} \\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{S}_{10} & \boldsymbol{S}_{11} \\ \boldsymbol{R}_{00} & \boldsymbol{R}_{01} & \boldsymbol{T}_{00} & \boldsymbol{T}_{01} \\ \boldsymbol{R}_{10} & \boldsymbol{R}_{11} & \boldsymbol{T}_{10} & \boldsymbol{T}_{11} \end{bmatrix} \begin{bmatrix} h_0(v) \\ h_1(v) \\ \overline{h}_0(v) \\ \overline{h}_1(v) \end{bmatrix} \quad (1)$$

ここで，h_0, h_1, $\overline{h}_0$, $\overline{h}_1$はそれぞれ次のように定義される3次多項式で，**混ぜ合せ関数**(blending function)とよばれる。

$$\begin{cases} h_0(t)=2t^3-3t^2+1 \\ h_1(t)=-2t^3+3t^2=h_0(1-t) \\ \overline{h}_0(t)=t^3-2t^2+t \\ \overline{h}_1(t)=t^3-t^2=-\overline{h}_0(1-t) \end{cases} \quad (2)$$

● Bézierパッチ

Bézierパッチ $\boldsymbol{H}(u, v)$, $0 \leq u \leq 1$, $0 \leq v \leq 1$ は，16個の制御点 $\boldsymbol{M}_{ij}$に基づいて次のように定義される曲面である($i, j=0, 1, 2, 3$)。

$$\boldsymbol{H}(u, v)=[w_0(u)\ w_1(u)\ w_2(u)\ w_3(u)] \times \begin{bmatrix} \boldsymbol{M}_{00} & \boldsymbol{M}_{01} & \boldsymbol{M}_{02} & \boldsymbol{M}_{03} \\ \boldsymbol{M}_{10} & \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13} \\ \boldsymbol{M}_{20} & \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23} \\ \boldsymbol{M}_{30} & \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33} \end{bmatrix} \begin{bmatrix} w_0(v) \\ w_1(v) \\ w_2(v) \\ w_3(v) \end{bmatrix} \quad (3)$$

ここで，重み関数 w_0, …, w_3は，それぞれ次のように定義される3次多項式とする。

$$\begin{cases} w_0(t)=(1-t)^3 \\ w_1(t)=3t(1-t)^2 \\ w_2(t)=3t^2(1-t) \\ w_3(t)=t^3 \end{cases} \quad (4)$$

法線ベクトルの計算

式(1)や式(3)のようなパラメトリックに表現された曲面上の点 $\boldsymbol{F}(u, v)=[f(u, v), g(u, v), h(u, v)]$ で，この曲面に対する法線ベクトルを求めるときには，まずこの曲面の接点ベクトルを2本求め，そ

*おおの　よしお　慶応義塾大学　情報科学研究所　〒223　神奈川県横浜市港北区日吉4-1-1

のどちらの接点ベクトルにも直交するような方向として法線ベクトルを決定することができる。

具体的には，$\boldsymbol{F}(u, v)$からuを微少量増やした方向への接線ベクトル$\frac{\partial}{\partial u}\boldsymbol{F}(u, v)$と，同様に$v$方向への接線ベクトル$\frac{\partial}{\partial v}\boldsymbol{F}(u, v)$とから，この2つの接線ベクトルの外積$\frac{\partial}{\partial u}\boldsymbol{F}(u, v)\times\frac{\partial}{\partial v}\boldsymbol{F}(u, v)$を求めればよい（ただし，2つの接線ベクトルのうちどちらかがゼロベクトルになってしまうような場合には，別の方向への接線ベクトルを求める必要がある）。

双3次パッチ$\boldsymbol{G}(u, v)$，Bézierパッチ$\boldsymbol{H}(u, v)$の場合には，それぞれ次のようになる。

$$\frac{\partial}{\partial u}\boldsymbol{G}(u, v)$$

$$=\frac{\partial}{\partial u}[h_0(u)\ h_1(u)\ \overline{h}_0(u)\ \overline{h}_1(u)]$$

$$\times\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{S}_{00} & \boldsymbol{S}_{01}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{S}_{10} & \boldsymbol{S}_{11}\\ \boldsymbol{R}_{00} & \boldsymbol{R}_{01} & \boldsymbol{T}_{00} & \boldsymbol{T}_{01}\\ \boldsymbol{R}_{10} & \boldsymbol{R}_{11} & \boldsymbol{T}_{10} & \boldsymbol{T}_{11}\end{bmatrix}\begin{bmatrix}h_0(v)\\ h_1(v)\\ \overline{h}_0(v)\\ \overline{h}_1(v)\end{bmatrix}$$

$$=[6u(u-1)\ \ 6u(-u+1)\ \ (3u-1)(u-1)\ \ u(3u-2)]$$

$$\times\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{S}_{00} & \boldsymbol{S}_{01}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{S}_{10} & \boldsymbol{S}_{11}\\ \boldsymbol{R}_{00} & \boldsymbol{R}_{01} & \boldsymbol{T}_{00} & \boldsymbol{T}_{01}\\ \boldsymbol{R}_{10} & \boldsymbol{R}_{11} & \boldsymbol{T}_{10} & \boldsymbol{T}_{11}\end{bmatrix}\begin{bmatrix}h_0(v)\\ h_1(v)\\ \overline{h}_0(v)\\ \overline{h}_1(v)\end{bmatrix},\qquad(5)$$

$$\frac{\partial}{\partial v}\boldsymbol{G}(u, v)$$

$$=[h_0(u)\ h_1(u)\ \overline{h}_0(u)\ \overline{h}_1(u)]$$

$$\times\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{S}_{00} & \boldsymbol{S}_{01}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{S}_{10} & \boldsymbol{S}_{11}\\ \boldsymbol{R}_{00} & \boldsymbol{R}_{01} & \boldsymbol{T}_{00} & \boldsymbol{T}_{01}\\ \boldsymbol{R}_{10} & \boldsymbol{R}_{11} & \boldsymbol{T}_{10} & \boldsymbol{T}_{11}\end{bmatrix}\begin{bmatrix}6v(v-1)\\ 6v(-v+1)\\ (3v-1)(v-1)\\ v(3v-2)\end{bmatrix},\qquad(6)$$

$$\frac{\partial}{\partial u}\boldsymbol{H}(u, v)$$

$$=\frac{\partial}{\partial u}[w_0(u)\ w_1(u)\ w_2(u)\ w_3(u)]$$

$$\times\begin{bmatrix}\boldsymbol{M}_{00} & \boldsymbol{M}_{01} & \boldsymbol{M}_{02} & \boldsymbol{M}_{03}\\ \boldsymbol{M}_{10} & \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13}\\ \boldsymbol{M}_{20} & \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23}\\ \boldsymbol{M}_{30} & \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33}\end{bmatrix}\begin{bmatrix}w_0(v)\\ w_1(v)\\ w_2(v)\\ w_3(v)\end{bmatrix}$$

$$=3[-(u-1)^2\ (3u-1)(u-1)\ \ u^2(1-3u)\ \ u^2]$$

$$\times\begin{bmatrix}\boldsymbol{M}_{00} & \boldsymbol{M}_{01} & \boldsymbol{M}_{02} & \boldsymbol{M}_{03}\\ \boldsymbol{M}_{10} & \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13}\\ \boldsymbol{M}_{20} & \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23}\\ \boldsymbol{M}_{30} & \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33}\end{bmatrix}\begin{bmatrix}w_0(v)\\ w_1(v)\\ w_2(v)\\ w_3(v)\end{bmatrix},\qquad(7)$$

$$\frac{\partial}{\partial v}\boldsymbol{H}(u, v)$$

$$=[w_0(u)\ w_1(u)\ w_2(u)\ w_3(u)]$$

$$\times\begin{bmatrix}\boldsymbol{M}_{00} & \boldsymbol{M}_{01} & \boldsymbol{M}_{02} & \boldsymbol{M}_{03}\\ \boldsymbol{M}_{10} & \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \boldsymbol{M}_{13}\\ \boldsymbol{M}_{20} & \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \boldsymbol{M}_{23}\\ \boldsymbol{M}_{30} & \boldsymbol{M}_{31} & \boldsymbol{M}_{32} & \boldsymbol{M}_{33}\end{bmatrix}$$

$$\times\begin{bmatrix}-(v-1)^2\\ (3v-1)(v-1)\\ v^2(1-3v)\\ v^2\end{bmatrix}\qquad(8)$$

光線と曲面パッチの交点

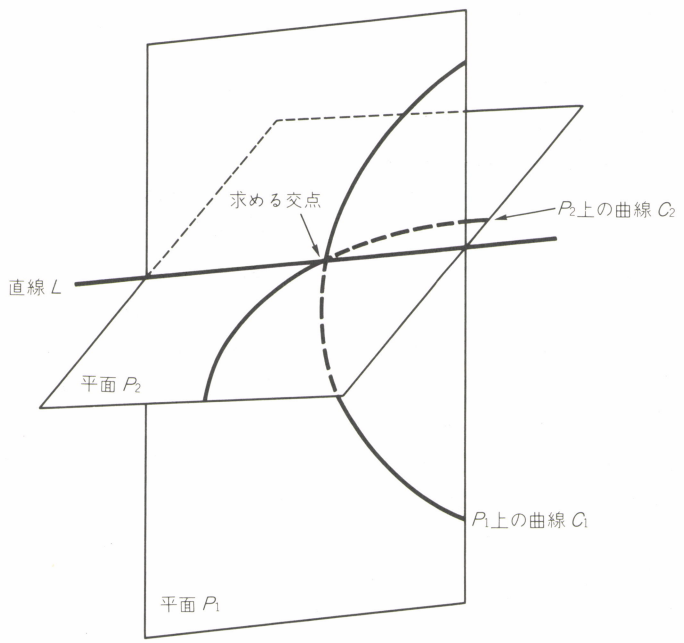
レイ・トレーシングでは，描画を行うために光線と物体との交点の計算を非常に多数回繰り返す必要がある。ところが前回にも述べたように，パラメトリックな曲面パッチの場合には，光線，すなわち直線との交点を求める直接的な手段がない。これまでにとられた方法は次の2種類に大別できる。

(1) 交点計算が直接できるような形に問題を変形し，数値計算の手法を利用して解く。

(2) 曲面パッチを，直線との交点を計算しやすい，例えば球などで包んでおき，その球と直線との交点の有無を調べる。もし交点があれば，パッチを細分化して各小パッチを包む球を求め，同様の計算を繰り返す。

一般的にいって，第2のアプローチの方がわかりやすいが，交点の位置が求まるまでに時間がかかる。第1のアプローチでは，数値計算の理論や手法に関する知識が必要であるが，計算速度を向上できる可能性が強い。

本連載では，まず前者の代表的なアルゴリズムとしてJ.T.Kajiyaの方法[2]を紹介したい。



● C_1は曲面パッチ F と平面 P_1 との交線
● C_2は曲面パッチ F と平面 P_2 との交線

図1　パッチと直線との交点

Kajiya のアルゴリズム

前回にも述べたように，双3次パッチも Bézier パッチも，それぞれの式を展開することによって容易に次の形に変換することができる。

$$\boldsymbol{F}(u,\ v)$$
$$=[u^3\ u^2\ u\ 1]\begin{bmatrix}\boldsymbol{A}_{33} & \boldsymbol{A}_{32} & \boldsymbol{A}_{31} & \boldsymbol{A}_{30}\\ \boldsymbol{A}_{23} & \boldsymbol{A}_{22} & \boldsymbol{A}_{21} & \boldsymbol{A}_{20}\\ \boldsymbol{A}_{13} & \boldsymbol{A}_{12} & \boldsymbol{A}_{11} & \boldsymbol{A}_{10}\\ \boldsymbol{A}_{03} & \boldsymbol{A}_{02} & \boldsymbol{A}_{01} & \boldsymbol{A}_{00}\end{bmatrix}\begin{bmatrix}v^3\\ v^2\\ v\\ 1\end{bmatrix}$$
$$=[u^3\ u^2\ u\ 1]\boldsymbol{A}\begin{bmatrix}v^3\\ v^2\\ v\\ 1\end{bmatrix}$$
$$=\sum_{i=0}^{3}\sum_{j=0}^{3}\boldsymbol{A}_{ij}u^iv^j \qquad (9)$$

計算上は，これが最も簡単な形なので，以後はこの形で考えることにする。

■問題の変形

Kajiya のアルゴリズムでは，直線 L と式(9)の曲面パッチ F との交点を直接に求めるのではなく，いったんこの問題を3次元空間中の2本の曲線の交点を求める問題に帰着させてから解く。そのため，直線 L を2枚の平面 P_1, P_2の交線として表現する。すると，F と L との交点は，F, P_1, P_2の3者の交点であり，それは F と P_1の交わり C_1と，F と P_2の交わり C_2との交点である。いうまでもなく，C_1はパッチ F と平面 P_1との交線であり，C_2はパッチ F と平面 P_2との交線であるので，この2本の空間曲線の交点を求めれば，最初の問題が解決できたことになる(**図1**)。

平面 P_1を

$$l_x^{(1)}x+l_y^{(1)}y+l_z^{(1)}z+l_w^{(1)}=0$$

すなわち，

$$[x\ y\ z\ 1]\begin{bmatrix}l_x^{(1)}\\ l_y^{(1)}\\ l_z^{(1)}\\ l_w^{(1)}\end{bmatrix}=[x\ y\ z\ 1]\ \boldsymbol{L}^{(1)} \qquad (10)$$

で表し，同様にして平面 P_2を

$$[x\ y\ z\ 1]\begin{bmatrix}l_x^{(2)}\\ l_y^{(2)}\\ l_z^{(2)}\\ l_w^{(2)}\end{bmatrix}=[x\ y\ z\ 1]\ \boldsymbol{L}^{(2)} \qquad (11)$$

で表現することにする。このようにすると，曲線 C_1は式(9)の各ベクトル $\boldsymbol{A}_{ij}$が同次座標で

$$\boldsymbol{A}_{ij}=[A_x{}^{(ij)}\ A_y{}^{(ij)}\ A_z{}^{(ij)}\ A_w{}^{(ij)}]$$

として表現されているものとすれば(i, j=0, 1, 2, 3)，

$$\sum_{i=0}^{3}\sum_{j=0}^{3}\boldsymbol{A}_{ij}u^iv^j\cdot\boldsymbol{L}^{(1)}=0 \qquad (12)$$

の解として表される。この式で

$$a_{ij}=[A_x{}^{(ij)}\ A_y{}^{(ij)}\ A_z{}^{(ij)}\ A_w{}^{(ij)}]\begin{bmatrix}l_x^{(1)}\\ l_y^{(1)}\\ l_z^{(1)}\\ l_w^{(1)}\end{bmatrix}$$

とおけば，式(12)は

$$\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}u^iv^j=0 \qquad (13)$$

の形に書けるので，これを解いて例えば u を v の関数で表すことができれば，それを式(9)に代入して曲線 C_1が求まることになる。同様にして，

$$b_{ij}=[A_x{}^{(ij)}\ A_y{}^{(ij)}\ A_z{}^{(ij)}\ A_w{}^{(ij)}]\begin{bmatrix}l_x^{(2)}\\ l_y^{(2)}\\ l_z^{(2)}\\ l_w^{(2)}\end{bmatrix}$$

とおけば，曲線 C_2は

$$\sum_{i=0}^{3}\sum_{j=0}^{3}b_{ij}u^iv^j=0 \qquad (14)$$

の解として表現できる。したがって，この2本の曲線 C_1, C_2の交点は式(13)と式(14)の共通解，すなわち，この2本の式を連立方程式として，その解 $u=\mu$, $v=\nu$ を求めればよい。最後にこの解を式(9)に代入すれば，曲面パッチ F と直線 L との交点の座標が求まることになる。

■終結式

式(13)と式(14)は未知数2個の連立方程式であるが，これを解く前に，未知数1個の連立方程式を解くことを考える。

$a(u)$ を u に関する m 次多項式とし，$a(u)=0$ の解を α_1, α_2, …, α_mとすると，次のように書ける。

$$a(u)=\sum_{i=0}^{m}a_iu^{m-i}$$
$$=a_0u^m+a_1u^{m-1}+\cdots+a_{m-1}u+a_m$$
$$=a_0(u-\alpha_1)(u-\alpha_2)\cdots(u-\alpha_m) \qquad (15)$$

同様にして，$b(u)$ を u に関する n 次多項式とし，$b(u)=0$ の解を β_1, β_2, …, β_nとして，次のように書く。

$$b(u)=\sum_{j=0}^{n}b_ju^{n-j}$$
$$=b_ou^n+b_1u^{n-1}+\cdots+b_{n-1}u+b_n$$
$$=b_0(u-\beta_1)(u-\beta_2)\cdots(u-\beta_n) \qquad (16)$$

このとき，$a(u)$ と $b(u)$ の**終結式**(resultant) $R(a,\ b)$ を次のように定義する。

$$R(a,b)=a_0^nb_0^m(-1)^{m+n-1}\prod_{\substack{i=1,\cdots,m\\ j=1,\cdots,n}}(\alpha_i-\beta_j) \qquad (17)$$

終結式に関しては次の性質が成り立つ(例えば文献5) の p.163, p.227 参照)。

「2つの方程式 $a(u)=0$ と $b(u)=0$ に共通の解が存在するための必要十分条件は，その終結式 $R(a,\ b)$ について $R(a,b)=0$ が成り立つことである。」

終結式 $R(a,\ b)$ は $\alpha_1, \alpha_2, \cdots, \alpha_m$や $\beta_1, \beta_2, \cdots, \beta_n$の値を知らなくても(すなわち，方程式 $a(u)=0$ や $b(u)=0$ を解かなくても)，それぞれの多項式の係数 a_0, a_1, …, a_m, b_0, b_1, …, b_nを用いて次のように書くことができる。

$$R(a,\ b)=\begin{vmatrix}a_0 & a_1 & \cdots & a_m & & \\ & a_0 & a_1 & \cdots & a_m & \\ & & \cdots & \cdots & \cdots & \cdots \\ & & & a_0 & a_1 & \cdots & a_m \\ b_0 & b_1 & \cdots & b_n & & \\ & b_0 & b_1 & \cdots & b_n & \\ & & \cdots & \cdots & \cdots & \cdots \\ & & & b_0 & b_1 & \cdots & b_n\end{vmatrix}\begin{matrix}\left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}n\text{行}\\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}m\text{行}\end{matrix} \qquad (18)$$

もちろん空所は0を意味する。

特に，現在われわれに関心があるのは $m=n=3$, すなわち $a(u)$も $b(u)$も3次多項式である場合，つまり $a(u)=a_0u^3+a_1u^2+a_2u+a_3$, $b(u)=b_0u^3+b_1u^2+b_2u+b_3$と書ける場合である。このとき式(18)は次の形になる。

$$R(a,\ b)=\begin{vmatrix}a_0 & a_1 & a_2 & a_3 & & \\ & a_0 & a_1 & a_2 & a_3 & \\ & & a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 & & \\ & b_0 & b_1 & b_2 & b_3 & \\ & & b_0 & b_1 & b_2 & b_3\end{vmatrix} \qquad (19)$$

この値が0であるかどうかを調べれば，$a(u)=0$ と $b(u)=0$ が共通解をもつかどうかがわかるわけである。

Kajiya は終結式の計算に，よく知られた式(19)でなく，より計算に便利な**ベズー**

の行列式形式(Bezout determinantal form)とよばれる次の形式を採用した。

$$R(a,\ b)=\begin{vmatrix} \begin{vmatrix} b_0 & b_1 \\ a_0 & a_1 \end{vmatrix} & \begin{vmatrix} b_0 & b_2 \\ a_0 & a_2 \end{vmatrix} & \begin{vmatrix} b_0 & b_3 \\ a_0 & a_3 \end{vmatrix} \\ \begin{vmatrix} b_0 & b_2 \\ a_0 & a_2 \end{vmatrix} & \begin{vmatrix} b_0 & b_3 \\ a_0 & a_3 \end{vmatrix}+\begin{vmatrix} b_1 & b_2 \\ a_1 & a_2 \end{vmatrix} & \begin{vmatrix} b_1 & b_3 \\ a_1 & a_3 \end{vmatrix} \\ \begin{vmatrix} b_0 & b_3 \\ a_0 & a_3 \end{vmatrix} & \begin{vmatrix} b_1 & b_3 \\ a_1 & a_3 \end{vmatrix} & \begin{vmatrix} b_2 & b_3 \\ a_2 & a_3 \end{vmatrix} \end{vmatrix} \tag{20}$$

■交点の計算

前項で $a(u)$, $b(u)$がともに u の3次多項式であるとき，方程式$a(u)=0$, $b(u)=0$が共通解をもつための必要十分条件が，式(20)で定義される終結式$R(a, b)$に関して，$R(a, b)=0$ が成立することであることがわかった。

ところで，われわれが解きたい問題で $a(u)=0$, $b(u)=0$ に相当するものは式(13)，式(14)であり，これらは実は u, v に関する双3次の多項式であった。そこで，式(13)を

$$\sum_{j=0}^{3}\left\{\left(\sum_{i=0}^{3}a_{ij}u^i\right)v^j\right\}=0$$

すなわち，

$$\begin{aligned}&(a_{33}u^3+a_{23}u^2+a_{13}u+a_{03})\,v^3\\&+(a_{32}u^3+a_{22}u^2+a_{12}u+a_{02})\,v^2\\&+(a_{31}u^3+a_{21}u^2+a_{11}u+a_{01})\,v\\&+(a_{30}u^3+a_{20}u^2+a_{10}u+a_{00})=0\end{aligned}\tag{21}$$

の形をしているものと考えることにする。ここで

$$a_j(u)=\sum_{i=0}^{3}a_{ij}u^i$$

すなわち，

$$\begin{aligned}a_3(u)&=a_{33}u^3+a_{23}u^2+a_{13}u+a_{03}\\a_2(u)&=a_{32}u^3+a_{22}u^2+a_{12}u+a_{02}\\a_1(u)&=a_{31}u^3+a_{21}u^2+a_{11}u+a_{01}\\a_0(u)&=a_{30}u^3+a_{20}u^2+a_{10}u+a_{00}\end{aligned}$$

とおけば，式(13)は次の形で表すことができる。

$$a_3(u)\cdot v^3+a_2(u)\cdot v^2+a_1(u)\cdot v+a_0(u)=0\tag{22}$$

同様に式(14)は

$$b_j(u)=\sum_{i=0}^{3}b_{ij}u^i$$

とおくことによって

$$\begin{aligned}&\sum_{j=0}^{3}\left\{\left(\sum_{i=0}^{3}b_{ij}u^i\right)v^j\right\}\\&=(b_{33}u^3+b_{23}u^2+b_{13}u+b_{03})\,v^3\\&+(b_{32}u^3+b_{22}u^2+b_{12}u+b_{02})\,v^2\\&+(b_{31}u^3+b_{21}u^2+b_{11}u+b_{01})\,v\\&+(b_{30}u^3+b_{20}u^2+b_{10}u+b_{00})\\&=b_3(u)\cdot v^3+b_2(u)\cdot v^2+b_1(u)\cdot v+b_0(u)\\&=0\end{aligned}\tag{23}$$

の形とみなすことができる。これらの式で係数として現れる u の3次多項式$a_i(u)$や $b_i(u)$を式(20)の a_iや b_iの代わりに使用すれば，終結式 $R(a, b)$が $a_i(u)$や $b_i(u)$の6次式，すなわち u に関する18次多項式となり(それを $r(u)$と書く)，方程式 $r(u)=0$ を解けば解 μ_1, μ_2, …, μ_qが求まる。これがはじめの曲面パッチ F と直線 L との交点が存在する u の値である。それぞれの $u=\mu_i$に対する v の値 ν_{ij}を決定して実際の交点の位置を求めようというのがKajiyaのアイデアである。

このアルゴリズムを実現するためには，少なくとも次の2つの問題を解決しなければならない。

1) 方程式 $r(u)=0$ をどのようにして解くか。
2) 解 $u=\mu_i$からどのようにして v の値を求めるか。

■高次代数方程式の解法

18次方程式 $r(u)=0$ を解くにあたってKajiyaは，ニュートン法に似てニュートン法よりも安定な**ラゲール**(Laguerre)の方法を採用した。ラゲールの方法の誘導は，例えば文献4)のpp.380-383に述べられている。この方法では，解の適当な推定値 ξ_0から出発し，次の式によって ξ_1, ξ_2, …を解に収束させる。

$$\xi_{k+1}=\xi_k-\frac{nr(\xi_k)}{r'(\xi_k)\pm\sqrt{H(\xi_k)}}\tag{24}$$

ここで n は多項式 $r(u)$の次数(この場合には18)であり，分母の平方根につく符号は $r'(\xi_k)$と合わせる。また，$H(\xi)$は次のように定義される。

$$H(\xi)=(n-1)[(n-1)r'(\xi)^2-nr(\xi)r''(\xi)]\tag{25}$$

解 μ_1が求まったら，$r(u)$を $u-\mu_1$で割れば，方程式の次数が1つ下がる。次は，その方程式に対して同様にラゲールの方法を適用すればよい。

$r(u)=0$ には18個の解があるが，この中でわれわれに意味があるのは，実根でしかも0から1までの範囲にあるものだけである。

●例

ラゲールの方法を使い，例えば $u^5-2=0$ の解を1つ求めてみよう。この場合の $r(u)$は5次式であるから $n=5$ である。また簡単な計算によって，次のことがわかる。

$$\begin{aligned}r(u)&=u^5-2\\r'(u)&=5u^4\\r''(u)&=20u^3\\H(u)&=800u^3\\\sqrt{H(u)}&=20u\sqrt{u}\end{aligned}$$

したがって，$r'(\xi_k)$は常に非負であり，反復のための式は次のようになる。

$$\xi_{k+1}=\xi_k-\frac{\xi_k^5-2}{\xi_k^4+4\xi_k\sqrt{2\xi_k}}\tag{26}$$

例えば，$\xi_0=1$ を初期値に選ぶと，次のようにわずか2ステップで解，すなわち2の5乗根が小数7桁まで求まる。

●ステップ1　$\xi_1=1.1502211$
●ステップ2　$\xi_2=1.1486984$
●ステップ3, …　$\xi_3=1.1486984$, …

■uからvを求めること

$r(u)=0$ の解 $u=\mu_k$が求まったら，それに対する v の値 ν_{k1}, ν_{k2}, …を求めなければならない。式(13)と式(14)に $u=\mu_k$を代入すると，v の連立3次方程式

$$\begin{cases}\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}{\mu^i}_k v^j=0\\\sum_{i=0}^{3}\sum_{j=0}^{3}b_{ij}{\mu^i}_k v^j=0\end{cases}\tag{27}$$

が得られる。これも前と同様に終結式を利用して解くこともできるが，Kajiyaの提案は，この2つの式の最大公約数を求めれば，もっと簡単に解けるというものである。

2つの多項式の最大公約数を求めるには，2つの整数の最大公約数を求めるための**ユークリッド**(Euclid)**の互除法**がそのまま使える。このアルゴリズムについては，例えば文献3)のp.251に詳しい解説がある。手順の概略は次の通りである。

●ユークリッドの互除法

2つの多項式を $a(u)$, $b(u)$とする。また，$a(u)$の係数の最大公約数を cont(a)と書き，$b(u)$の係数の最大公約数を cont(b)と書く。

・ステップ1　cont(a)と cont(b)の最大公約数を dとする。$a(u)\leftarrow a(u)/$cont(a)とし，同様に $b(u)\leftarrow b(u)/$cont(b)とする。
・ステップ2　$a(u)/b(u)$の余り(u)を

求める。$r(u)=0$ならステップ4にいく。$r(u)$が0でない定数なら$b(u)\leftarrow 1$としてステップ4にいく。

・ステップ3　$a(u)\leftarrow b(u)$, $b(u)\leftarrow r(u)/\mathrm{cont}(r)$とし，ステップ2に戻る。

・ステップ4　計算終了。$d\cdot b(u)$が求める最大公約数である。

●例

例えば，

$a(u)=(u-1)(u-2)(u-3)(u-4)=u^4-10u^3+35u^2-50u+24$,

$b(u)=(u-1)(u-2)(u+1)(u+2)=u^4-5u^2+4$

の場合を試してみよう。

・ステップ1　$\mathrm{cont}(a)=\mathrm{cont}(b)=1$であるから，$d\leftarrow 1$。また，$a(u)\leftarrow a(u)/1$, $b(u)\leftarrow b(u)/1$とする。

・ステップ2　$a(u)/b(u)$の余り$r(u)$は$-10u^3+40u^2-50u+20$である。$r(u)$は定数ではないのでステップ3にいく。

・ステップ3　$a(u)\leftarrow u^4-5u^2+4$とする。また，$\mathrm{cont}(r)=10$であるので，$b(u)\leftarrow -u^3+4u^2-5u+2$とする。

・ステップ2　$a(u)/b(u)$の余り$r(u)$は$6u^2-18u+12$である。$r(u)$は定数ではないので，ステップ3にいく。

・ステップ3　$a(u)\leftarrow -u^3+4u^2-5u+2$とする。また，$\mathrm{cont}(r)=6$であるので，$b(u)\leftarrow u^2-3u+2$とする。

・ステップ2　$a(u)/b(u)$はちょうど割り切れて，余り$r(u)$は0である。ループから抜け出してステップ4にいく。

・ステップ4　$d\cdot b(u)=u^2-3u+2=(u-1)(u-2)$が求める最大公約数である。

おわりに

今回は，曲面パッチのレイ・トレーシングを直接的な数値計算を利用して行うKajiyaのアルゴリズムについて，その手順の概要を解説した。しかしこのままでは，まだプログラムにするにはいろいろな問題がある。次回はその点について考えてみたい。

参考文献

1) 伊理正夫：「数値計算」, 理工系基礎の数学12, 朝倉書店, 1981

2) James T.Kajiya：“Ray Tracing Parametric Patches”, SIGGRAPH'82 Proceedings, pp.245-254

3) Donald E.Knuth：“Seminumerical Algorithms”, The Art of Computer Programming, Vol.2, Addison-Wesley, 1981(邦訳 中川圭介：「準数値算法/算術演算」サイエンス社, 1986)

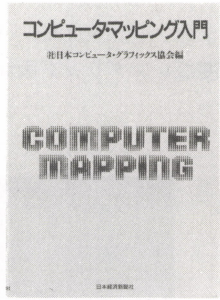
4) Anthony Ralston, Philip Rabinowitz：“A First Course in Numerical Analysis”, McGraw-Hill, 1978

5) 新数学事典, 大阪書籍, 1979

BIBLIOGRAPHY

コンピュータ・マッピング入門

編／(社)日本コンピュータ・グラフィックス協会
発行所／日本経済新聞社
☎03(270)0251
初版発行日／昭和63年11月4日
定価／2,800円
判型／B5判
ページ数／218ページ



今，コンピュータ・マッピングとか地図情報処理と総称されている分野が大変注目されている。

そもそもコンピュータ・マッピングとは何なのだろうか。そして一体どのように活用したらよいものなのか。まだ特殊で非常に新しい分野であり，またその技術も日進月歩という状況である。

本書は，コンピュータ・マッピングの入門用として書かれたものである。編集にあたった日本コンピュータ・グラフィックス協会は，コンピュータ・グラフィックスの普及活動などを目的とした社団法人である。そしてこの協会の分科会の一つにマッピング分科会がある。

マッピング分科会では，コンピュータ・マッピングに関する情報交換，研究発表，動向調査，NICOGRAPHの準備などの活動をしているが，ここでの議論に教育の問題があった。コンピュータ・マッピングの急速な普及に対し，教育の遅れが目立ってきたのである。

日本では，大学でコンピュータ・マッピングに関する専門教育を行ってないため，この分野に携わる企業では技術者を社内で養成する必要が生じている。また，地方自治体などでもシステムが導入されだしたため，ユーザー教育が大きな課題の一つとなってきている。

本書は，現時点での地図情報処理を理論と応用の両面から解説し，1冊にまとめ上げた初の本格的入門書である。地図情報処理にすでに携わっている方にとってより一層理解を深めるために，そして初心者の方にとってはまたとない優れた入門書である。

本書は15章からなり，各章は次の通り。◎イントロダクション◎地図の基礎知識◎コンピュータ・マッピングの歴史◎コンピュータ・マッピングの定義と分類◎コンピュータ・マッピングのシステム構成◎ハードウエア◎図形記述方法◎データベース◎空間解析◎地理情報システム(GIS) ◎施設管理◎ナビゲーション◎AMデジタル地図の作成◎標準化◎コンピュータ・マッピングの将来

注目の新製品　3次元レンダリングを高速処理するディスプレイ COMTEC 3555 レンダリング・ターボ

テクスチャ＆バンプ・マッピングをはじめ，従来ソフトウエアで処理していた高度なレンダリング機能をハードウエア化した3次元CGの概要と，特徴について述べる。

増田　孝*

はじめに

ダイキン工業は，電子機器事業に参入して以来，特に3次元コンピュータ・グラフィックス（CG）の分野において斬新な製品を次々と開発，3次元グラフィック・ディスプレイのパイオニアとしての役割を果たしてきた。特に，COMTEC DS300シリーズはリアルタイムに3次元グラフィックの処理を実現する優れた性能・機能を有し，ローカル3次元を実用化するディスプレイとの評価を得，3次元CGの需要を喚起してきた。

しかし，3次元アプリケーションのニーズはサイエンティフィック・ビジュアライゼーション（Scientific Visualization），高度なレンダリング画像によるプレゼンテーションというようにますます高度化し，グラフィック・ディスプレイに対しても，複雑な3次元モデルを“より高速に，もっとリアルに”というハードな要求がさらに強まってきている。

当社では“Reality&Realtime”を3次元CG開発の基本思想とし，常にグラフィック・アプリケーションの要望を先取りする高性能・高機能ディスプレイの開発を目指している。そしてこのほど超高速・高機能グラフィック・ディスプレイCOMTEC3555をベースに，レンダリング機能を大幅に強化したCOMTEC3555RT（Rendering Turbo）を開発した。

COMTEC3555RTは，テクスチャ・マッピング（Texture Mapping），バンプ・マッピング（Bump Mapping），トランスペアレンシィ（Transparency：透明感表示）などのレンダリング機能や曲面プリミティブなどのモデリング機能を国産機として初めてハードウエアでもち，以上の機能によるリアルな3次元画像を，簡単な操作でスピーディーに表示することを可能にした。

昨年（1988年）8月にアトランタ（Atlanta）で行われたSIGGRAPH'88に参考出展したところ，その高度な3次元機能が非常に高い評価を得た。そしてさらに最終的なブラッシュアップを経て，11月に正式にリリースされた（**図1**）。

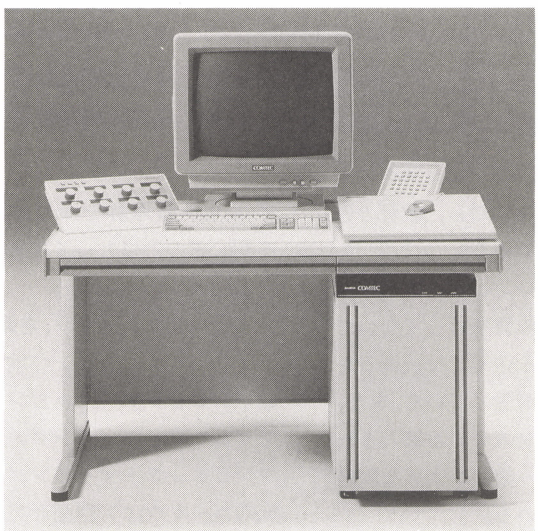


図　1

製品の特徴

COMTEC3555RTは，超高速シェーディング（Gouraud Shading）処理，フォン（Phong）シェーディング機能，ローカル・ライティング機能などを搭載したCOMTEC3555に，上述のレンダリング&モデリング機能を追加した新製品である。以上の機能を実現したことにより，機械設計のみならず意匠設計までを含めた本格的なトータル3次元CADや，工業製品・建築景観分野のダイナミックなデザイン・シミュレーションが可能になる。設計・デザイン環境を一変するパワーを秘めたスーパー・レンダリング・マシンとして，その威力が発揮されるものと自負している。

■テクスチャ・マッピング

テクスチャ・マッピングは，図柄・模様などの2次元画像を3次元物体の表面に張り付けるレンダリング技術である。

*ますだ　たかし　　ダイキン工業㈱　電子機器事業部企画課　〒163　東京都新宿区西新宿2-6-1

◉ 図 3〜5, 12 は COLOR IMAGES 参照。

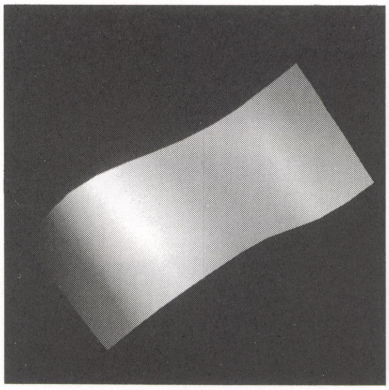


図 6　線織面

図 7　フィレット曲面

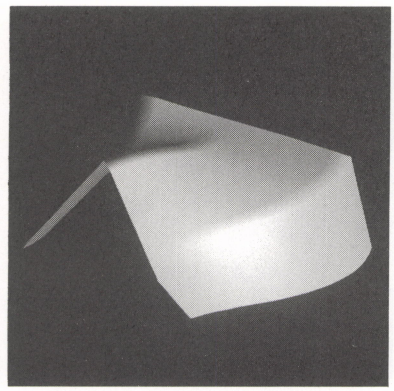


図 8　スイープ曲面

COMTEC3555RT では，イメージ・スキャナなどで読み込んだ画像パターンをディスプレイのもつ専用メモリに描き込み，ポリゴンや任意曲面で構成された 3 次元モデルに対して高速にマッピングを行う。

具体的な方法を以下に示す。まず，テクスチャ・データをディスプレイ内部のエミュレーション・プロセッサから VME バスを介し，フォン・シェーディング・プロセッサ上の 1,024×1,024 ピクセルのマッピング・メモリに描き込む。その際，ユーザーはメモリの U（横方向）と V（縦方向）のアドレスを指定することにより，メモリ上の任意の位置にデータを登録することができる。

3 次元モデルにマッピングするときは，モデルのどの部分にテクスチャを張り付けたいかを，対応する頂点ごとにマッピング・メモリの U・V アドレスで指定すると，フォン・シェーディング・プロセッサがテクスチャ・データをモデルの各ピクセルについて張り付けていく（**図 2**）。

テクスチャ・マッピング機能は，フォン・シェーディング時に使用できる。COMTEC3555RT は，3 次元モデルを形成する領域内の各ピクセルごとに法線ベクトルを求め，このベクトルを用いて光源や視線の方向を考慮しながら輝度値を求める計算をするという，フォンのアルゴリズムを忠実に行いながら画像表示している（テクスチャ・マッピングをする際には，物体の色としてそのピクセルに対応するマッピング・メモリの色を使い，輝度計算を行う）。しかもフォン・シェーディングは 8 個のプロセッサで並列処理を行うため，リアルなシェーディングを施したマッピング画像を瞬時に表示することができる。

また，マッピング時にはローカル・ライティング機能も併用できるため，光源の種類（平行光源・点光源）や数（1 つのビューにつき最大 8 個），周囲の環境光の明るさが自由に設定できる。さらに，マッピングした結果に対する周辺の散乱光や，鏡面反射のシミュレーションも可能である。マッピングした 3 次元物体によりリアルな質感を与え，さらに現実に近い光の条件下で素早く表現することができる（**図 3**）。

従来この機能は，ホスト・コンピュータ上のソフトウエアで膨大な時間をかけて処理していたが，専用ハードウエアの開発によって，これまでの処理時間を 1/15〜1/30に短縮した（当社ベンチマークテストによる。DEC 社 VAX-11/750 比）。処理時間の大幅な短縮によって，マッピングしたモデルの高速表示，インタラクティブな視点の変更が可能となった。

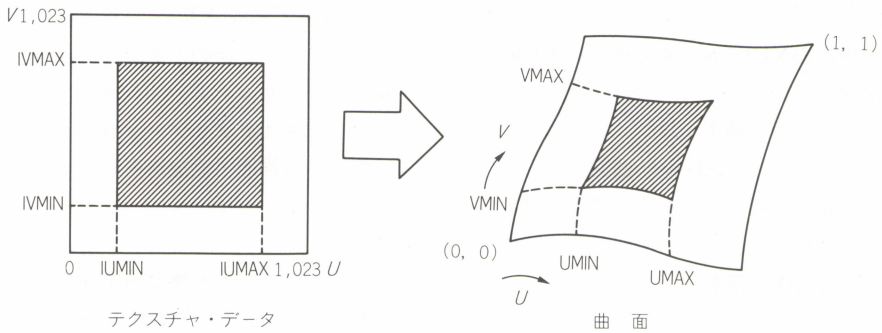


図　2

■バンプ・マッピング

バンプ・マッピングは滑らかな物体の表面に凹凸による陰影を発生させ，凹凸感のある表面を表示することにより，モデルの材質感を表現するマッピング技法である。

具体的な方法は，原則としてテクスチャ・マッピングと同じである。バンプ・データ（凹凸を表す 8 ビットのデータ）をフォン・シェーディング・プロセッサ上のマッピング・メモリに登録し，3 次元モデルのどの部分に対応させるかをマッ

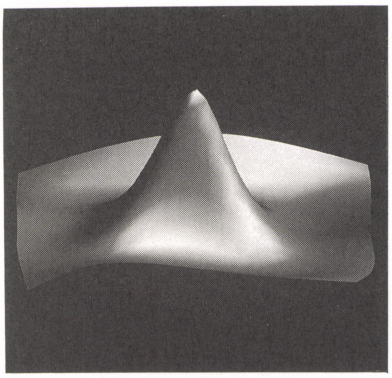


図9　カーブメッシュ曲面

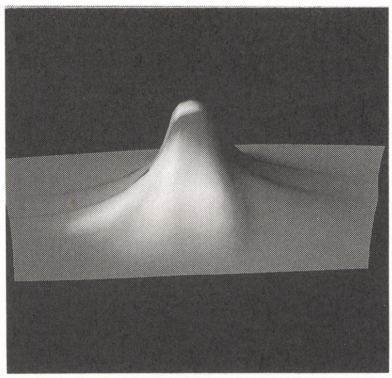


図10　ポイントメッシュ曲線

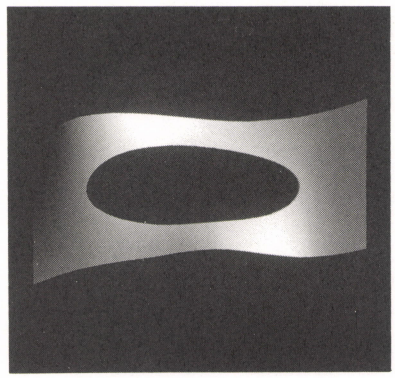


図11　トリム曲面

◉ 図6〜8は前ページ。

ピング・メモリの $U \cdot V$ アドレスで指定する。さらに，3次元モデルの法線ベクトルをどちらの方向へ振らせるかを決めるバンプ UV ベクトルを指定する。3次元モデルの各ピクセルに対応するバンプ・データを法線ベクトルを振らせる重み（振り幅）の係数として使用し，バンプ UV ベクトルに掛け合わせて求めたベクトルを各ピクセルの法線ベクトルに加えることにより，もとの法線ベクトルを振らせて表面の凹凸感を表現する。

テクスチャ・マッピングと同様，フォン・シェーディング時に利用が可能であり，ローカル・ライティング機能との併用ができる。さらにテクスチャ・マッピングも併せて使うことができる（**図4**）。また専用ハードウエアの開発により，従来と比べてテクスチャ・マッピングの項で述べたのと同等の処理時間の短縮を実現している。

■トランスペアレンシィ

フォン・シェーディング時に透明面の透明率を指定することによって，透明面とその背景の面の色を透明率の割合で合成して表示する。透明面の各ピクセルごとに面の角度を考慮しながら色と輝度を求めて表示していくため，従来のスクリーンドア方式による半透明表示以上のリアルな透明感が得られる。ガラスやプラスチックのような質感を表すのに最適である。また，テクスチャ・マッピングやバンプ・マッピングとの併用が可能なので，ステンドグラスやクリスタルイメージなど高度な3次元画像の表示が実現できる（**図5**）。

■曲面プリミティブ

自由曲面は平面や簡単な立体プリミティブ（球・円柱など）では表現できない微妙な曲面が生成できるものとして，最近とみに注目を集めている機能である。自動車の車体から家電製品にいたるまで，今日の工業製品・設計物で曲面形状を含まないものはほとんどないといっても過言ではない。それだけにそれらの製品の設計・デザイン評価などをCGで行う場合，曲面形状をいかに，簡単に短時間でリアルに表現できるかということが重要な問題となる。

COMTEC3555RTは自由曲面をプリミティブとしてハードウエアに搭載し，上述の問題の解決を図った。さらに，さまざまなパターンの曲面形状表示を実現するため，多種多様な曲面プリミティブを採用している。

1)　線織面

2本のNURBS（Non-Uniform Rational B-Spline）曲線の対応する点を直線で結んでできる曲面（**図6**）。

2)　フィレット（Fillet）曲面

中心のNURBS曲線と外周のNURBS曲線2本からできる円弧によって生成される曲面（**図7**）。

3)　スイープ（Sweep）曲面

断面を定義するキャラクタカーブの間を，ガイドカーブに沿って補間して作成される曲面（**図8**）。

4)　カーブメッシュ（Curvemesh）曲面

4端の曲線を補間するタイプのクーンズ（Coons）曲面（**図9**）。

5)　ポイントメッシュ（Pointmesh）曲面

制御点の座標，2つの接線ベクトル，ツイスト・ベクトルで指定される双3次のクーンズ曲面（**図10**）。

6)　トリム（Trim）曲面

トリミング機能により自由曲面の切取りや穴あけも可能である（上記のすべての曲面に対して指定可能，**図11**）。

その他，球・Revolution（NURBS曲線による回転体）などの立体プリミティブ，Bézier，Spline，B-Spline，NURBSなどの曲線プリミティブも装備している。

このような機能をローカル側でもつことにより，“データ量の低減によるホストCPU負荷の軽減と高速レスポンス”“微妙な曲面のリアルな表現”“モデリング作業時間の低減”など多くのメリットが得られる（**図12**）。

以上，RTの新機能を中心に説明した。この他にもテクスチャ・マッピングの項でふれたフォン・シェーディング，ローカル・ライティングをはじめ，超高速シェーディング処理，全方向高速ベクトル描画，1,670万色同時表示，高速セクショニング，32ビット浮動小数点による図形定義空間，16ビットZバッファなど，さまざまな3次元アプリケーション・ニーズに対応する性能・機能を有している。

問合せ先
ダイキン工業㈱　電子機器事業部
東京☎03(344)8103，大阪☎06(373)1201，
名古屋☎052(562)4727
※ショールームで随時デモンストレーションを行っている。

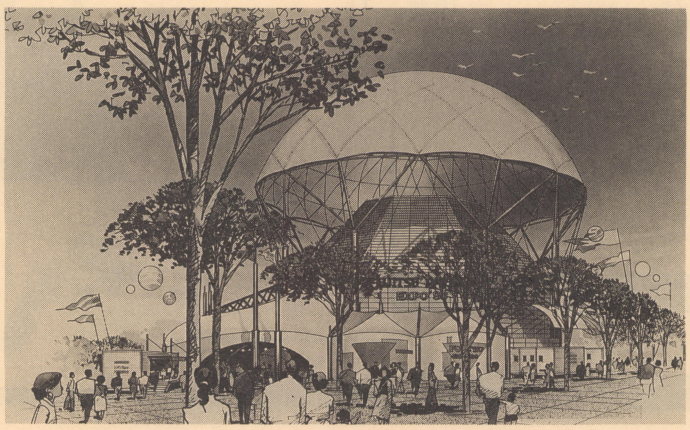
NEWS SCAN

富士通が「国際花と緑の博覧会」への出展計画を発表

富士通は1990年4月から大阪で開催される「国際花と緑の博覧会」への出展計画をまとめた。同社のパビリオンである「富士通パビリオン」では，カナダのアイマックス社の新全天周映写機による立体上映方式を用いた立体フルカラーのコンピュータ・グラフィックス（CG）映像が直径24mの全天周スクリーンに映写される。映像制作には，元ローレンス・リバモア研究所のネルソン・マックスを中心としたヒカリキネマ社と富士通とが共同であたる。なお，CGの制作には富士通製スーパー・コンピュータVPシリーズが用いられる。

映像は，分子生物学，生物組織学の最新の科学理論とデータをもとに制作されており，「植物の生命活動と動物の筋肉活動の神秘」を分子レベルで映像表現するという世界初の試みであると，同社では述べている。太陽のエネルギーが植物の中で光合成によって変換され，糖として蓄えられ，その植物を人間や動物が食べ，糖が血液によって筋肉細胞に運ばれてそこで分解され運動エネルギーに変換され，人間や動物が動くことができるというストーリーをCGと実写を織り混ぜて映像化している。

上映時間は約17分間で，収容人員は1回330名。期間中に約170万人の来館を見込んでいる。



富士通パビリオンの外観図

兼松エレクトロニクスが建築CAD展示会

兼松エレクトロニクスは，2月15日から17日まで東京・銀座で建築用CADシステムの展示会「第7回KEL建築コンピュータ展示会」を開催する。新バージョン「Gaudi-Super 3」の発表の他，他の建築CAD，CADセミナーを行う。

日時　1989年2月15日（水）～17日（金）

会場　銀座文祥堂イベントホール（東京都中央区）

問合せ先　兼松エレクトロニクス㈱　応用システム事業部CAD営業部

☎03(436)9621

'88国際画像機器展

第19回画像工学コンファレンスとの併催による'88国際画像機器展が，東京産業貿易会館で1988年12月7日から9日まで開催された。本展示会は今回で10回目を迎え，出展社数も120社とこの分野では最大の規模である。ブースはいずれも派手な飾りつけがなくデスクトップ・ディスプレイの展示のみだが，コンパニオンがいない分，中身が濃い充実した展示会という印象があった。今年は3日間で15,750人を集める盛況だった。

工場における製品の自動目視検査などをターゲットにした製品が比較的多く見受けられたが，今回は特にパーソナルコンピュータを用いた画像処理システムの多さが目立った。パソコン画像処理は，ワークステーション以上のコンピュータをベースにしたシステムよりも安価なことから，例えばTOSPIXの入力用にPC-9801を使えるような製品も登場していた。一方，入力機器としてはCCDカメラのオンパレードという感じで，この分野の技術進歩の速さが実感された。こうした周辺機器の低価格化はシステム自体の二極化，つまりほぼ定着した感のあるハイエンド画像処理システムと，コスト・パフォーマンス＋特化した機能をもつパソコン画像処理システムの分化に，一層拍車をかけることになるだろう。

日本IBMが3次元グラフィック・ライブラリHOOPSをリリース

日本アイ・ビー・エムはPS/55シリーズ用として，ファモティクが販売している3次元グラフィック・ライブラリHOOPSの国内販売を行うと発表した。

HOOPSは米国イサカ・ソフトウエア社が開発したもので，レンダリング機能とオブジェクト指向の3次元階層構造をもつソフトウェアである。米国ではVersaCAD，SOLUTION 3000，MDIなどのアプリケーションに採用されており，ソフトウエア開発者用のグラフィック・ツールとして使用されている。

GRADE用にアポロ・ワークステーションが年間10億円受注

日本アポロコンピュータは，日立造船情報システムが開発・販売している汎用CAD/CAMシステムGRADEシリーズ用のコンピュータとして，10億円分のワークステーションをOEM供給する契約を日立造船情報システムと締結した。この契約は1989年に発効するが，現在新規の引き合いだけで40件以上を超えているため，年内に累計で100台の受注が見込まれるとしている。中心となっているアポロDOMAINシリーズはDN 3500で，一部DN 4500も含まれている。

掲示板

IRIS²誕生

AIT

国内外のEWSで実績のある対話型統合CAEシステムIRISが，シリコングラフィックス社のGWS(IRIS-4D)に移植され，IRISダブルが誕生しました。3次元処理が迫られる今，AITではIRIS-4Dのジオメトリ・エンジンを使用し，より対話的でリアルタイムな3次元処理が可能なCAEシステムを実現しました。これにより，3次元処理に重点をおいた対話型統合CAEシステム“IRIS²”をユーザーに提供できるようになりました。

詳細につきましては，下記にお問い合わせ下さい。

問合せ先

㈱エイ・アイ・テクノロジー

〒101　東京都千代田区東神田2-6-2 タカラビル9F　☎03(864)6912

アークブレインよりのお知らせ

アークブレイン

アークブレインでは，C言語用グラフィック・ライブラリ「Ultra C Graph」の，最近新たに各社より発売されたCコンパイラへの対応を済ませました。動作確認を済ませたCコンパイラは以下の通りです。

- Microsoft Quick C Ver. 1. 1
- Microsoft C Ver. 5. 1（MS-DOS版）
- TURBO C Ver. 2. 0
- Lattice C Ver. 4. 0

「マドレールSF」，「Ultra C Graph」の販売店用デモプログラムを用意致しましたので，ぜひご利用下さい。

問合せ先

㈱アークブレイン

〒151　東京都渋谷区幡ヶ谷3-20-2-207

☎03(375)8968　FAX 03(375)8767

VENUSリリース

アステック

3次元ソリッド・モデラーVENUSは，東京大学精密機械工学科・木村研究室の協力を得て，当社とデザイン・オートメーションで開発してまいりましたが，いよいよ1月末より発売を開始することになりました。

ソリッド・モデラーは，真の3次元モデラーとしてこれまで多くの人に求められながらあまり普及しませんでした。それは，「応答性」「操作性」「価格」に問題があったためでした。VENUSはこれらの問題を解決し，次世代の3次元CADのプラットフォームとすることを目指しています。また，価格もロード・モジュールで85万円とし，ソリッド・モデラーとしては導入しやすい価格となっています。

教育・公的研究機関に対しては，アカデミック・プライスとして30万円で提供いたします。ぜひ3次元CADの教育用としてご検討下さい。操作が簡単なため，短い授業時間内でも十分操作を覚えられ，ソリッド・モデラーの概念を理解することができます。

今回のVENUSのリリースは，Sun-3，Sun-4用です。GMWウィンド・システム上で稼働します。その他の機種についてもいくつか予定していますが，ご希望がございましたらご連絡下さい。

問合せ先

㈱アステック

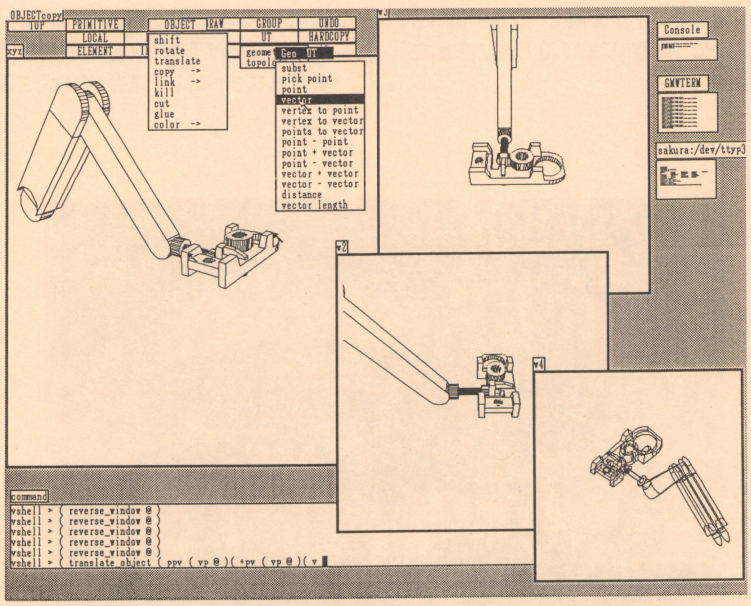
〒150　東京都渋谷区道玄坂1-22-12 長島第一ビル3F

☎03(477)1541　FAX 03(476)0183

強力なコンピューティング+高度なグラフィックス。アサヒステラGS1000は時代の先端ニーズと対話するところから生まれました。

旭テクノコンピュータ

アサヒステラGS1000は，革新的技術有するきわめて優れた頭脳集団として名高い米国ステラ社が生み出した，全く新



アステック／出力例

しいタイプのコンピュータです。スーパー・コンピュータとグラフィックスの統合をインタラクティブ環境で実現した画期的なシステムです。

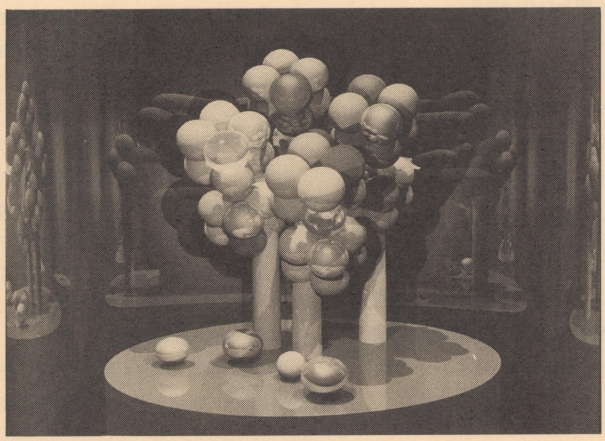
旭テクノコンピュータは，ステラ社が世界のお客様に提供する製品・教育・サポート・サービスに，旭化成グループならではの広範な科学技術分野での豊かな経験と実績とを生かしたノウハウを付加して，日本のお客様にお届けしていきます。

アサヒステラ GS1000 の並列処理アーキテクチャの心臓部ともいえる SPMP（シンクロナス・パイプライン・マルチプロセッサ）は，多重パラレル機能ユニットと広バンド幅の DataPath™機能とで構成されています。内蔵された 4 個のマルチプロセッサが 50ns 単位できわめて効率良く同期実行を行います。また，DataPath によってデータ転送速度は 3G バイト／秒以上の高速となっています。

グラフィック機能も際だった数値計算能力と同様に強力であり，完全に一体化されています。

より速く，より柔軟に，より美しく…時代の先端ニーズを満たすアサヒステラ GS1000 が，これからのコンピュータリゼーションをリードします。

問合せ先
旭テクノコンピュータ㈱　営業本部
☎ 03(507)2316



インテグラ／照明シミュレーション例

高度な照明シミュレーションを可能にするターボビームトレーシング

インテグラ

ターボビームトレーシング（TBT）は，完全大域照明モデル（Complete Global Illumination Model）を，インテグラ独自の双方向レイ・トレーシング法を用いて表現した，高度な照明シミュレーションを可能にする画像生成システムです。

TBT は，従来のレイ・トレーシング法による写真のような実在感をもつ画像生成機能に加え，照明シミュレーションには不可欠である拡張反射光と鏡面反射光のそれぞれについて光線の複雑な相互反射を追跡計算します。これにより，ラジオシティ法などの照明シミュレーションを高い精度で実現でき，その実行も双方向レイ・トレーシング法により非常に短時間で行うことができます。

問合せ先
㈱インテグラ
〒170　東京都豊島区駒込 2-7-11　比米ビル 2F　☎ 03(949)6620，6640

Sun-3 用リアルタイム画像処理ボード発売

インターニックス

MATROX 社は，すでに VMEbus 用リアルタイム画像処理ボード「MVP-VME」シリーズを販売しているが，今回新たにサン・マイクロシステムズ社の「Sun-3」ワークステーションにスロットインして使用できる，リアルタイム画像処理ボード「MVP-S」シリーズの販売を開始した。

MVP-S は，特定領域に対するウィンド処理，リアルタイム画像間演算，32 ビット・プレーンのフレーム・バッファを「Sun-3」上で提供している。さらに強力なグラフィックス機能を搭載して，テキスト，グラフィックスのオーバレイを高速に実行できる。また MC68000 をローカル CPU として搭載することによって，画像処理時のホスト CPU の負荷を大幅に軽減することが可能となった。ソフトウエア・ライブラリは，ハイレベルの画像処理コマンド「IMAGER」がファームウエアで提供され，MVP-S/LIB（エミュレーション・ソフト）との併用によって Sun OS 上でアプリケーション開発をサポートしている。

MVP-S は，自動検査，画像計測，位置決め，ロボット制御，形上認識などのマシンビジョン・アプリケーション，画質改善が求められる医用画像処理，ソナー画像処理，リモートセンシングといったアプリケーションにおいて，高機能リアルタイム・システムを「Sun-3」上で実現する。

問合せ先
インターニックス（株）
〒160　東京都新宿区西新宿 7-4-7
☎ 03(369)1438

編集後記

第1回 PIXEL CG グランプリ
応募締切り 1月20日㈮
多数の応募をお待ちしています。

第1回PIXEL CG グランプリはコンピュータ・グラフィックスを使用した静止画作品のコンテストで，プロ・アマおよび使用するハードウエアを問わずどなたでも応募できます。アートばかりでなく，産業シミュレーションやプレゼンテーションも応募できますので，幅広いアプリケーションからの作品をお待ちしています。(詳細は色紙の告知ページにあります)

◘CAD/CAMやコンピュータ・グラフィックスがより開かれたものになっていくためには，何か特別なものという印象ではなく，ごく普通のコンピュータ・システムとして認識していただかなくてはいけないもののようです。

現状では，やや専門的な知識，トレーニング，センス，業界（通常は村や部落とよばれています）での付合い，学会やグループへの加担など，まだまだ近代化されていないような部分が残されているようです。

CAD/CAMやコンピュータ・グラフィックスの担当者は，非常に特殊なシステムを扱っているというスペシャリスト意識のおかげで，苛酷な労働や低賃金（失礼，ごく一部の話だけですが）をものともせずに頑張ってこられたわけでしょうが。いつまでもスペシャルであっては，スペシャルということは数少ないということで，いけないわけです。

この分野は奥が深いですから，入門から奥義を極めるまでにはいくつかの階層があり，入門部分を開放したとしても，スペシャル意識をくすぐれる部分はまだまだ多いと思われます。

◉本誌88年12月号「NEW PRODUCTS」69ページの日本アイテック社の記事中に一部誤りがありました。TDI EXPLORE が IRIS-4D シリーズへの移植も進められる予定である。という部分ですが，すでに日本アイテック社では 4D シリーズへ移植を完了し，販売を行っています。関係者各位にお詫びして訂正をいたします。

編集スタッフ募集

PIXEL編集部では，誌面充実のため，編集スタッフ（正社員）を募集いたします。

〈応募資格〉
4年制大学を卒業した27歳位までの方で，コンピュータ・グラフィックスまたはコンピュータについての知識のある方。雑誌編集の経験は問いません。

〈仕事の内容〉
PIXELの取材および編集

〈応募方法〉
応募される方は，事前に連絡のうえ，履歴書を下記までご送付下さい。

図形処理情報センター　担当：河内
〒101　東京都千代田区神田神保町1-64
神保町協和ビル6F　☎03(293)6161

次号のご案内

89年3月号（2月20日発売）

大特集　保存版レンダリング，アニメーション，2Dシステム総覧

〈解説記事〉
- システムのいろいろと選び方
- ユーザーのための3次元システムの機能と使い方
- ユーザーのための2次元システムの機能と使い方
- 3次元アニメーションの作り方
- パーソナル CG システム

〈製品記事〉
- Wavefront　● ALIAS
- PERSONAL LINKS
- EMA, MARK-2D/3D, YUI
- image maker　● Intelligent Light
- monolith　● VISHAD　● RIO
- Super Tableau　● EXPLORE
- ChapReyes　● S-Products
- ANTICS, MOSLA
- MOVIE. BYU　● REALS
- RAY-TREK II　● ARTS
- TURBO BEAM TRACING
- 麗子，彩子　● PSY-CRONE
- C-TRACE　● FINEPAC
- ARTSTAR　● PictureMaker

（順不同）

◉弊社にて，バックナンバーや既刊別冊，また最新の別冊号「CAD/CAM,CG年鑑'89」の販売も行っています。ご利用下さい。

定期購読・別冊号のお申込み方法

本誌は毎月1日発行（前月20日発売）です。全国の書店で発売されますが，最寄りの書店にない場合は，直接購読をお勧めします。

定期購読料は送料込みで，
半年　5,800円／1年　11,500円

新しく定期購読を申し込まれる方（継続の方は下記をご参照下さい）は，とじ込みハガキをご利用下さい。購読料のお支払いは，雑誌とともに郵送する郵便振替用紙もしくは現金書留でお願いします。

本誌は'80年11月，『図形と画像』（季刊）として創刊され，'83年1月，隔月刊となり『PIXEL』と改題されました。'84年1月から月刊発行です。

バックナンバー，別冊号は次の通り。

■『図形と画像』('81/秋～'82/冬。創刊号，'81/春，夏は売切れ）定価1,200円 送料250円
■『PIXEL』('83/1-2～11-12，'84/1月号～）
定価　980円　送料　95円
■別冊『アプリケーション』
定価 4,500円　送料　無料
■別冊『グラフィックスガイド』
定価 3,300円　送料　無料
■別冊『CAD/CAM/CAE の基礎』
定価 2,800円　送料　無料
■別冊『CAD/CAM, CG 総覧'88』
定価 3,500円　送料　無料

定期購読継続のお申込み方法

本号で定期購読が切れた方は，巻末ハガキを利用して継続の手続きをして下さい。その際，必ず購読者番号をご記入下さい。

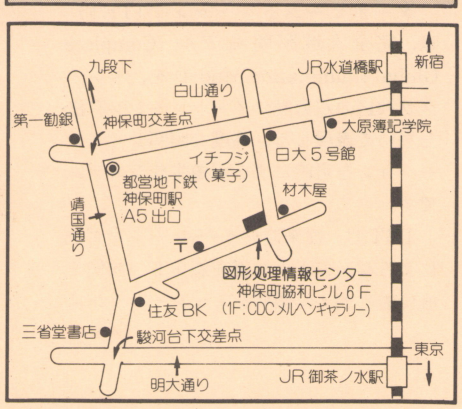
〈申込み先：販売部〉

広告のお申込み方法

本誌に広告掲載を検討されている方は，広告媒体資料をお送りいたします。

〈申込み先：広告部〉

お申込みは FAX でもお受けいたしております。
FAX：03(293)6164



PIXEL　89／2月号

1989年2月1日発行　No.77
編集・発行人　河内　隆幸

発行所　図形処理情報センター
〒101　東京都千代田区神田神保町1-64
神保町協和ビル6階　電話 03(293)6161(代表)
発売元　オーク出版サービス　電話 03(291)7031
印刷所　千代田平版社
定　価　980円(送料　95円)
郵便振替　東京　5-50653

© 1989 図形処理情報センター

郵便はがき

料金受取人払

神田局承認

5463

差出有効期間
1990年11月
24日まで

切手不要

101-00

010

千代田区神田神保町
一―六四
神保町協和ビル

図形処理情報センター
読者サービス係行

会社名
(学校名)

所属
部課名

上記の
住所

〒　　電話

フリガナ
ご氏名　　年令　　男・女

○印をおつけ下さい。

業種　コンピュータ・メーカー，図形処理機器メーカー，ソフト会社，
システム会社，輸送用機器，一般産業用機械，工作機械，精密機械，
重電，弱電，建築・土木，繊維，地図，機械部品，化学プラント，
エンジニアリング，印刷・製版，デザイン，アニメーション，
マスコミ，学校（教員，学生），コンサルタント，その他（　　）

業務　設計，研究・開発，技術管理，営業技術，コンピュータ室，生産，
ソフト開発，デザイン，経営者，技術，その他（　　）

郵便はがき

料金受取人払

神田局承認

5463

差出有効期間
1990年11月
24日まで

切手不要

101-00

010

千代田区神田神保町
一―六四
神保町協和ビル

図形処理情報センター
読者サービス係行

会社名
(学校名)

所属
部課名

上記の
住所

〒　　電話

フリガナ
ご氏名　　年令　　男・女

○印をおつけ下さい。

業種　コンピュータ・メーカー，図形処理機器メーカー，ソフト会社，
システム会社，輸送用機器，一般産業用機械，工作機械，精密機械，
重電，弱電，建築・土木，繊維，地図，機械部品，化学プラント，
エンジニアリング，印刷・製版，デザイン，アニメーション，
マスコミ，学校（教員，学生），コンサルタント，その他（　　）

業務　設計，研究・開発，技術管理，営業技術，コンピュータ室，生産，
ソフト開発，デザイン，経営者，技術，その他（　　）

PIXEL 89/2月号

PIXEL 資料請求

ご入用な広告ページの資料請求番号を○でお囲み下さい

表2 表3 表4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38

PIXEL アンケート

・本号でおもしろかった記事のタイトルを２つご記入下さい
　１）
　２）
・本号でおもしろくなかった記事のタイトルを２つご記入下さい
　１）
　２）
・本誌に対する要望，感想，批判など自由にお書き下さい

・本誌のお買い求め方法は　直接購読／書店
・当センターからのDMは　届いている／届いていないので送ってほしい／不要

PIXEL 定期購読・別冊号申込

請求書：要／不要

□本号で定期購読がきれましたので '89年３月号より　半年／１年間　継続を申し込みます／中止します　（購読者No.　　）
□新規に定期購読を '89年３月号より　半年／１年間　申し込みます
□バックナンバーを申し込みます

「PIXEL」
'89/1,
'88/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
'87/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

別冊
2号「アプリケーション」
4号「グラフィックスガイド」
5号「CAD/CAM/CAEの基礎」
CAD/CAMアルファNo.1／No.2

□別冊7号「CAD/CAM, CG年鑑89」を申し込みます。□別冊6号「CAD/CAM, CG総覧88」を申し込みます。

お送り先（葉書のおもて面と同じ場合は不要）

住所　〒

社名・部署

氏名　　電話

PIXEL 89/2月号

PIXEL 資料請求

ご入用な広告ページの資料請求番号を○でお囲み下さい

表2 表3 表4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38

PIXEL アンケート

・本号でおもしろかった記事のタイトルを２つご記入下さい
　１）
　２）
・本号でおもしろくなかった記事のタイトルを２つご記入下さい
　１）
　２）
・本誌に対する要望，感想，批判など自由にお書き下さい

・本誌のお買い求め方法は　直接購読／書店
・当センターからのDMは　届いている／届いていないので送ってほしい／不要

PIXEL 定期購読・別冊号申込

請求書：要／不要

□本号で定期購読がきれましたので '89年３月号より　半年／１年間　継続を申し込みます／中止します　（購読者No.　　）
□新規に定期購読を '89年３月号より　半年／１年間　申し込みます
□バックナンバーを申し込みます

「PIXEL」
'89/1,
'88/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
'87/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

別冊
2号「アプリケーション」
4号「グラフィックスガイド」
5号「CAD/CAM/CAEの基礎」
CAD/CAMアルファNo.1／No.2

□別冊7号「CAD/CAM, CG年鑑89」を申し込みます。□別冊6号「CAD/CAM, CG総覧88」を申し込みます。

お送り先（葉書のおもて面と同じ場合は不要）

住所　〒

社名・部署

氏名　　電話

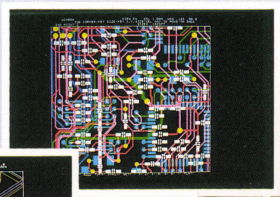
CADAM®

それはCAD/CAMシステムの代名詞です。

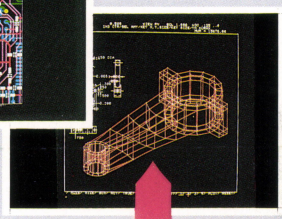
そしていま、CADAMは大きく発展しています。

●特定の
設計技術分野

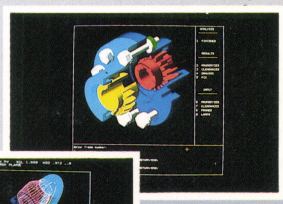
●IPC(電子回路設計)



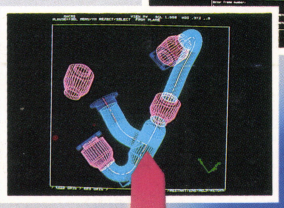
●FEM

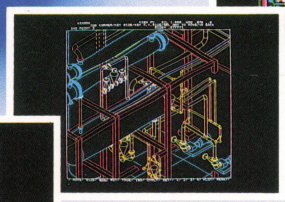


●ISD（対話型ソリッド設計）

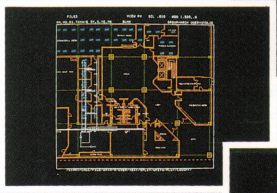


●3次元対話
モジュール

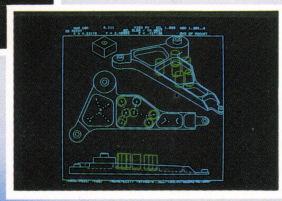




●3次元配管プラント



●AEC



●NC-Ⅱ

複合アプリケーション・
パッケージ

高度モデリング・ツール

●CAE
概念設計
解析／
シミュレーション

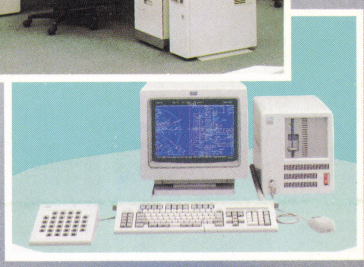
CADAM

基本対話モジュール
CADAMファイル

ファミリー・シリーズ

●IBM6150
5080装置

●PROFESSIONAL
CADAM

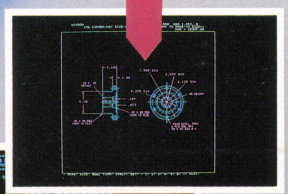


●IBM5550シリーズ
IBM-PC/AT

●MICRO CADAM

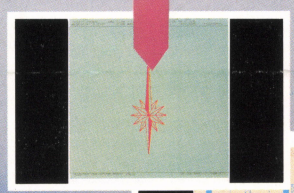
●事務／
生産管理との結合
●CAMへの
指令

CADAM ACCESS

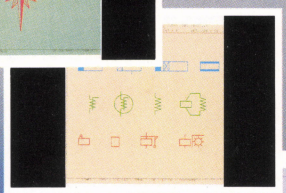


●マクロ
ジオメトリー

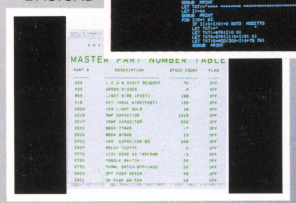
CADAMライブラリー



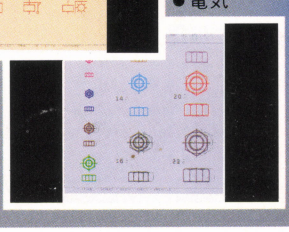
●建築



●BASICAD



●電気



●IUE

●ユーザー・
カストマイズド・
ツール
●パラメトリック設計

●CADAM
ネットワーキング

●機械

CADAM
SYSTEMS COMPANY

キャダムシステム株式会社

本　　社　〒141　東京都品川区大崎3丁目5番2号　エステージ大崎5F
☎(03)491-5648(代)　　Fax.(03)491-5720

大阪営業所　〒530　大阪市北区芝田1丁目4番14号　芝田町ビル3F
☎(06)376-0408(代)　　Fax.(06)376-1027

★カタログ請求、お問合せは
左記までどうぞ

人材募集
CAD/CAM経験者
（35歳以下）募集中
お問合せ先：本社総務部

※CADAM®は、CADAM Inc. の登録商標です。

（資料請求番号　表3）

PIXEL
1989年2月1日発行/毎月1回1日発行/通巻77号
1984年5月2日第3種郵便物認可

図形処理情報センター
〒101 東京都千代田区神田神保町1-64
神保町協和ビル　電話 03(293)6161

たしかな技術をシステムに——E&Mの神鋼電機
エレクトロニクス メカトロニクス

彩先端CHC
カラー ハード コピー

フレームバッファSPI-3と組合わせたCHC

300dpiで高精細プリント
しかもA3を55秒で出力する
高速・鮮明カラーハードコピー

CAD/CAM、CGの画像処理など色々な分野で活躍している神鋼電機のカラーハードコピーCHCシリーズ。300dpiの高解像度・高速タイプも各種シリーズ化し、ひときわ高精細なプリントが、ハイスピードでプリントアウトできます。しかもパソコン用のプリンタドライバなどにより、いっそう簡単に出力できるようになりました。

■98シリーズの出力が簡単になりました！
NEC PC98/9801シリーズに追装するだけで手軽にCRT画面のハードコピーが取れる専用ボードが登場。セントロニクスを介し無調整で動作でき、市販されているほとんどのソフトに対応できます。

■27万色のスーパーマルチカラーも可能！
さらに上位コンピュータなどのビデオ信号を短時間で取込むとともに、各種の画像処理機能を内蔵した「フレームバッファSPI-3」を組合わせれば、最高27万色の超多色表現など、より美しい処理ができます。

神鋼 カラーハードコピー
神鋼電機 SHINKO ELECTRIC CO.,LTD.
A3用 A4用 B6用 CHC シリーズ

■お問い合せは一神鋼電機株式会社　端末機器本部へ
●本　社／東京都中央区日本橋3-12-2 朝日ビル
〒103 ☎(03)274-1129
●大 阪 支 社／大阪市東区北浜3-1 大阪グリーンビル
〒541 ☎(06)203-2244
●名古屋支社／名古屋市中村区名駅4-6-18 名古屋ビル
〒450 ☎(052)581-8544

定価　980円
（資料請求番号 表4）
雑誌　07651-2

PIXEL

コンピュータ・グラフィックスとCAD/CAM/CAE雑誌ピクセル

新しいCG表現を可能にするメタボール

89

2

図形処理情報センター